



---

UNIVERSITÉ DE QUÉBEC À MONTRÉAL

ARCHITECTURE DE SIMULATION MULTIRESSOURCE POUR UN RÉSEAU GRILLE  
BASÉ SUR OBS

MÉMOIRE  
PRÉSENTÉ  
COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN INFORMATIQUE

PAR  
MARYEM OURTI

DÉCEMBRE 2009

## UNIVERSITÉ DU QUÉBEC À MONTRÉAL

Service des bibliothèques

### Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement n°8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

## **DÉDICACE**

À ma maman et mon papa

## REMERCIEMENTS

Je remercie tout d'abord, ma directrice de recherche Halima Elbiaze, d'avoir pris de son temps pour m'aider et m'encourager et je la remercie surtout pour ses conseils et son appui tout au long de ce projet.

Je remercie mes parents d'avoir cru en moi, de m'avoir encouragé tout au long de mon cursus et d'avoir sacrifié autant pour mon éducation et mes études.

Je remercie également tous les membres du laboratoire de téléinformatique et réseaux (LTIR) de l'UQAM, et tout particulièrement Martin et Heykel pour leur soutien.

J'adresse mes plus sincères remerciements à tous mes proches et mes amis qui m'ont toujours soutenue et encouragée au cours de mes études.

Enfin, je tiens à remercier toutes les personnes qui m'ont apporté leur soutien et aide et qui ont contribué à l'élaboration de ce mémoire.

## TABLE DES MATIERES

LISTE DES FIGURES .....	vi
LISTE DES SIGLES .....	vii
RÉSUMÉ .....	viii
CHAPITRE I : INTRODUCTION .....	1
1.1 PROBLEMATIQUE.....	2
1.2 CONTRIBUTION .....	3
1.3 STRUCTURE DU MEMOIRE .....	3
CHAPITRE II : INTRODUCTION AUX RESEAUX OBS ET AUX GRILLES.....	4
2.1 INTRODUCTION AUX RESEAUX OBS .....	4
2.1.1 ARCHITECTURE GENERALE D'OBS:.....	6
2.1.2 CONCEPTS DE BASE.....	7
2.1.2.1 Agrégation des bursts .....	7
2.1.2.2 Routage et assignement de longueur d'onde (RWA).....	10
2.1.3 SIGNALISATION.....	10
2.1.3.1 TAG (Tell-And-Go).....	11
2.1.3.2 JET (Just-Enough-Time).....	11
2.1.3.3 JIT (Just-In-Time).....	12
2.1.3.4 Horizon .....	13
2.1.3.5 WR-OBS.....	14
2.2 GRILLES DE CALCUL : .....	15
2.2.1 INTRODUCTION .....	15
2.2.2 ARCHITECTURE GENERALE DE LA GRILLE .....	17
Fabrique.....	17
Connectivité.....	18
Resource .....	18
Collective.....	19
Applications.....	19
CHAPITRE III : GRILLE BASEE SUR OBS .....	20

3.1 INTRODUCTION .....	20
3.2 ARCHITECTURE D'UN RESEAU GOBS : .....	21
3.2.1 NŒUD PERIPHERIQUE .....	23
3.2.1.1 Classification des tâches de la grille .....	24
3.2.1.2 Coordination entre les utilisateurs/ ressources de la grille et le réseau .....	24
3.2.2 LES ROUTEURS INTERMEDIAIRES .....	27
3.3 SIGNALISATION : .....	27
3.3.1 JET (JUST ENOUGH TIME): .....	28
3.3.2 JIT (JUST IN TIME): .....	29
3.3.3 HORIZON .....	31
3.3.4 WR-GOBS (WAVELENGTH ROUTED-GOBS) .....	32
3.4 ROUTAGE ANYCAST: .....	33
CHAPITRE IV : ARCHITCTURE DES SIMULATIONS ET RESULTATS OBTENUS .....	36
4.1 INTRODUCTION .....	36
4.2 MECANISME D'ALLOCATION ET RESERVATION DES RESSOURCES .....	38
4.3 SIMULATIONS ET RESULTATS .....	39
4.3.1 EVALUATION DE LA PERFORMANCE D'INFRASTRUCTURE ACTIVE POUR GOBS .....	40
4.3.1.1 Environnement de simulation .....	41
4.3.1.2 Résultats et analyses : .....	43
4.3.2 ARCHITECTURE MULTI-RESSOURCE POUR GOBS .....	46
4.3.2.1 Environnement de simulation .....	47
4.3.2.2 Résultats et analyses .....	48
CONCLUSION .....	53
RÉFÉRENCES .....	54

## LISTE DES FIGURES

Figure 1: Composants d'un réseau optique .....	5
Figure 2: Architecture d'un réseau OBS .....	7
Figure 3: Agrégation des bursts ( <i>Burst Assembly</i> ) .....	8
Figure 4: Burst Disassembly .....	8
Figure 4: Tell and Go (TAG) .....	11
Figure 5: Just-Enough-Time (JET).....	12
Figure 6: Just-In-Time (JIT).....	13
Figure 7: Horizon .....	14
Figure 8: WR-OBS.....	15
Figure 9: Architecture de la grille et modèle OSI [23].....	17
Figure 10: Architecture de d'un réseau GOBS.....	22
Figure 11: Infrastructure d'un réseau GOBS [40].....	25
Figure 12: Interface réseau pour les utilisateurs de la grille (GUNI) .....	26
Figure 13: JET-GOBS .....	29
Figure 14: JIT-GOBS .....	30
Figure 15: BBS (Bimodal Burst Switching).....	32
Figure 16: WR-GOBS .....	33
Figure 17: Architecture d'un routeur actif [50] .....	37
Figure 18: structure d'un burst actif et Non-actif .....	38
Figure 19: Demande de ressources et transmission de la tâche basée sur Jet.....	39
Figure 20: Réseau NSFNET .....	42
Figure 21: Topologie2.....	42
Figure 22: NSFNET: Taux de perte de bursts vs. Charge du réseau .....	44
Figure 23: NSFNET: Délai de transmission vs. Charge du réseau.....	44
Figure 24: Topologie2: Taux de perte vs. Charge du réseau .....	45
Figure 25: Topologie2: Délai de bout-en-bout vs. Charge du réseau .....	46
Figure 27: NSFNET: Taux de perte vs. Nombre de nœuds ressources .....	48
Figure 28: NSFNET: Taux de réussite vs. Nombre de nœuds ressources .....	49
Figure 29: Topologie2: Taux de perte vs. Nombre de nœuds ressources.....	50
Figure 30: Topologie2: Taux de réussite vs. Nombre de nœuds ressources.....	50
Figure 31: Taux de perte vs. Charge .....	51
Figure 32: Taux de réussite global vs. Charge .....	52

## LISTE DES SIGLES

ACK	Acquittement (Acknowledge en anglais)
BBS	Bimodal Burst Switching
FDL	Fiber Delay Line
GOBS	Grille de Calcul basées sur OBS
GRNI	Grid Resource Network Interface
GUNI	Grid User Network Interface
IP	Internet Protocol
JET	Just Enough Time
JIT	Just In Time
MPLS	Multi-Protocol Label Switching
NACK	Acquittement négatif
NP	Network Processor
OBS	Optical Burst Switching
OCS	Optical Circuit Switching
O-E-O	Optique-Electronique-Optique
OXC	Optical Cross Connect
OPS	Optical Packet Switching
QoS	Quality of Service
RWA	Routing and Wavelength Assignment
TAG	Tell and Go
TCP	Transmission Control Protocol
WDM	Wavelength Division Multiplexing
WR	Wavelength Routed



## RÉSUMÉ

Le concept de la grille de calcul a émergé récemment comme infrastructure apte à répondre aux exigences des applications scientifiques et économiques en termes de stockage et calcul. Le principe de la grille est de regrouper toute la puissance inexploitée des ordinateurs d'un réseau pour la mettre à la disposition des applications nécessitant de grands calculs. Il s'agit également d'utiliser l'espace de stockage inutilisé des serveurs pour y entreposer des données. Une grille de calcul représente donc une collection de ressources informatiques réparties en plusieurs emplacements et agissent en tant que ressources de traitement unifiées ou comme une collection d'ordinateurs virtuels. L'avantage de la grille consiste en un traitement plus rapide et plus efficace. Elle permet aussi d'éliminer l'inconvénient de lier étroitement des machines spécifiques à des travaux spécifiques. Puisqu'elle permet à des utilisateurs et à des applications d'accéder et de contrôler efficacement diverses ressources de différentes locations distribuées.

Toutefois, sans un moyen de transport de données assez rapide, il y aurait aucun avantage dans le traitement distribué des données. La commutation optique par rafales (OBS) a été identifiée comme la technologie idéale pour répondre aux exigences de la grille. Cette approche est connue sous le nom de grille basée sur OBS « GOBS ». Dans ce mémoire nous étudions et évaluons la performance d'un réseau GOBS. Nous présentons aussi le problème de multi-ressource dans le cas où plusieurs ressources informatiques seront dédiées pour le traitement des tâches de la grille. Les résultats obtenus nous ont permis de démontrer que l'approche multi-ressource est efficace pour le traitement distribuée au sein de la grille et particulièrement dans le cas de grandes tâches. Elle permet de réduire le taux de perte et aussi d'assurer un traitement plus rapide.

**Mots clé :** Grille de calcul, GOBS, OBS, tâche, Burst, Rafale, Paquet de contrôle, ressource.

# CHAPITRE I

## INTRODUCTION

Malgré les progrès importants concernant la performance des systèmes de calcul et de stockage au cours de la dernière décennie, il existe toujours un grand nombre de problèmes qui ne peuvent pas être résolus dans un délai limité ni avec un superordinateur ou un cluster. Plusieurs exemples émergent dans divers domaines de recherches tels que l'astrophysique, la météorologie, etc. [5] [4]. D'où le besoin de tirer profit de la croissance explosive des systèmes informatiques interconnectés via Internet et d'employer cette infrastructure comme un seul système virtuel afin de résoudre ces problèmes.

Ceci a mené la notion de *grille*. Une grille représente donc une collection de ressources informatiques réparties en plusieurs emplacements et agissent en tant que ressources de traitement unifiées ou comme une collection d'ordinateurs virtuels. Ainsi, la distribution géographique des ressources permet une gestion décentralisée sans avoir besoin d'un point d'administration central.

Selon les domaines d'application et le scénario de son utilisation, différents types de grille ont été identifiés :

- Les *grilles de calculs* qui combinent un grand nombre de ressources de traitement afin d'offrir une exécution rapide et des résultats performants.
- Les *grilles de données* qui sont orientées vers la gestion des données distribuées, en supportant non seulement les fonctions de stockage mais également le regroupement et l'analyse des données de différentes sources.
- Les *grilles de services* qui couvrent plus ou moins tous les réseaux grilles n'appartenant pas aux deux classes précédentes et qui désignent parfois les grilles qui offrent un traitement de services en temps réel.

Cependant, sans un moyen de transport de données assez rapide, il y aurait aucun avantage dans le traitement de données distribuées. Les réseaux optiques représentent donc le candidat parfait pour les grilles de calcul, puisque ils utilisent la technologie *WDM (Wavelength Division Multiplexing)* qui consiste à multiplexer différents types de trafic dans un même canal optique permettant ainsi d'offrir des bandes passantes très larges.

De plus, l'introduction des techniques de commutation permet non seulement d'augmenter le rendement mais aussi la possibilité de s'adapter à une large variété de caractéristiques du trafic. Parmi ces techniques de commutation on trouve *OBS (Optical Burst Switching)* appelé aussi *Commutation par rafales* [13] qui représente une solution prometteuse pour répondre aux exigences des grilles. Puisque son format de transport peut satisfaire les demandes des utilisateurs en termes de bande passante et fournir une utilisation efficace des ressources réseau. Ainsi, la fusion des deux technologies vise à atteindre plusieurs objectifs et pose de défis de recherche dans le domaine des télécommunications.

## 1.1 Problématique

Plusieurs recherches ont été menées dans le domaine des grilles optiques. Les principaux aspects abordés sont : l'architecture et la conception générale des grilles optiques, les algorithmes d'optimisation pour l'ordonnancement du trafic et le routage, et la gestion des ressources du réseau. Les premiers travaux se sont intéressés principalement à utiliser la technique de *Commutation par circuit (OCS)* comme infrastructure réseau pour les grilles. Beaucoup de progrès ont été accomplis récemment en testant le fonctionnement des grilles basées sur OCS pour l'exécution de diverses applications scientifiques, notamment. Par contre et malgré le peu de travaux traitant la fusion des grilles et la commutation par rafales, OBS a été identifié comme étant la technologie appropriée aux grilles.

Le défi principal dans les grilles basées OBS est celui de savoir comment implémenter une infrastructure de réseau efficace et intelligente pour satisfaire les demandes des utilisateurs et maximiser l'utilisation des ressources. La solution réside donc à identifier les algorithmes de routage et d'allocation des ressources appropriés. La plupart des modèles proposés se concentrent sur un seul problème soit l'acheminement des données ou l'assignement des ressources grilles.

Dans ce contexte, nous nous sommes posés les questions suivantes : a) à partir d'une architecture basée sur des routeurs offrant l'ordonnancement et/ou la découverte des ressources dans une grille

(appelé aussi *Routeurs actifs*), comment peut-on améliorer la performance de l'allocation des ressources réseau, b) comment subdiviser des tâches qui nécessitent un grand nombre de ressources et les affecter aux ressources disponibles dans le réseau et ainsi améliorer l'allocation de ces ressources, c) peut-on prouver cette approche par des simulations et étudier les performances de ces algorithmes.

## 1.2 Contribution

Dans ce mémoire, nous présenterons une nouvelle approche intégrant l'utilisation de routeur actif et qui étudie le problème de multi-ressource au sein d'un réseau GOBS. Plusieurs ressources informatiques seront dédiées donc pour le traitement des tâches de la grille. Notre objectif est de réduire au minimum le taux de perte qui résulte de la contention dans le réseau OBS et de l'indisponibilité des ressources de la grille. Pour cela notre travail sera structuré comme suit :

1. La première étape consiste à évaluer l'intégration de routeur actif dans un réseau GOBS. Cette étude nous permettra de voir l'impact de l'utilisation simultanée des deux types de trafic : actif et non actif.
2. La deuxième étape consiste à étudier l'approche multi-ressource. Dans ce contexte, nous nous sommes intéressés uniquement aux trafics actifs. Nous évaluons l'impact de cette proposition en faisant varier le nombre de ressources, l'emplacement des nœuds (routeur actif et ressource), et la subdivision des tâches.

## 1.3 Structure du mémoire

La suite de ce mémoire s'articule comme suit : le deuxième chapitre nous introduisons les réseaux optiques à commutation par rafales (OBS). Nous consacrons également, dans le cadre de ce chapitre, une partie pour introduire les grilles de calcul. Le chapitre 3 présente l'état de l'art sur les travaux qui étudient les réseaux grille basées sur OBS. Dans le chapitre 4, nous présenterons l'architecture GOBS étudiée et les résultats des simulations.

## CHAPITRE II

### INTRODUCTION AUX RESAUX OBS ET AUX GRILLES

#### 2.1 Introduction aux réseaux OBS

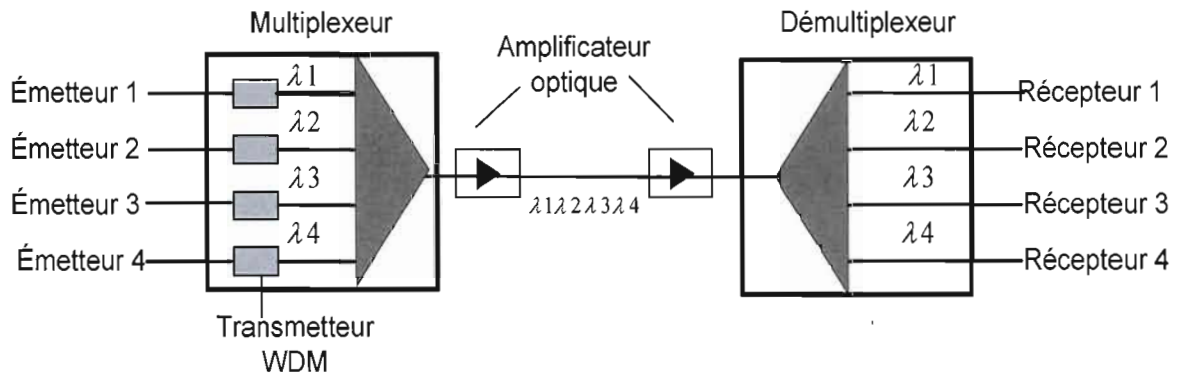
L'évolution des communications de données par fibres optiques est provenue de la nécessité de l'augmentation de la bande passante dans différents types de réseaux. Cette augmentation est due à la croissance exponentielle des demandes d'utilisation d'Internet, des applications de calcul, des applications multimédia, des applications à temps réel, etc. Les premiers réseaux optiques à grande capacité sont apparus au cours des années 90.

Les réseaux optiques sont basés sur l'utilisation de multiples longueurs d'ondes. Les composants réseaux (routeurs optiques) déterminent comment les longueurs d'onde seront transmises, assemblées et mises sur le réseau. L'intégration de la technologie WDM (Wavelength Division Multiplexing) a ajouté une capacité additionnelle aux fibres optiques existantes et qui peut atteindre jusqu'à 10 Gb/s. La technologie WDM [37] consiste à envoyer simultanément dans la même fibre optique plusieurs signaux numériques à la même vitesse de modulation, chacun sur une longueur d'onde distincte. L'exploitation de cette propriété augmente la quantité de données transportées sur une fibre et donc multiplie la capacité du réseau.

La première génération de réseaux optiques exigeait une conversion « optique à électrique à optique » (OEO) à chaque lien entre deux nœuds. Cependant, le passage à une transmission « tout-optique » était la clef du succès des réseaux optiques de la deuxième génération. Ce qui offre une transparence, un haut débit, et grande capacité de traitement de plusieurs trafics en parallèle.

Les composants qui assurent un transport tout optique sont les convertisseurs de longueurs d'onde (*Wavelength Converters*), les multiplexeurs de longueurs d'onde (*Wavelength Add/Drop Multiplexers*) et les amplificateurs. Les convertisseurs se chargent d'associer à chaque signal

entrant une longueur d'onde qui va le représenter dans le réseau optique et, en sortie, réalisent l'opération inverse. Le multiplexeur regroupe plusieurs longueurs d'onde pour une transmission sur une seule fibre optique. Le démultiplexeur exécute la fonction inverse. Les amplificateurs permettent de régénérer un signal optique faible en le convertissant en un signal électronique puis l'amplifier et le reconvertir en un signal optique.



**Figure 1:** Composants d'un réseau optique

L'évolution des réseaux tout optique a commencé avec les réseaux à commutation de circuit qui consiste à établir une connexion entre deux nœuds tout au long de la durée de la transmission des données. Ceci garantit l'allocation de canaux réservés et assurer ainsi la transmission des données tout en respectant une certaine qualité de service. Cependant, l'inconvénient principal des réseaux OCS est le nombre limité de longueurs d'onde par fibre. Puisque dans un grand réseau et à cause de cette faiblesse, il serait impossible d'établir des connexions entre tous les nœuds du réseau, ce qui aurait pour effet d'augmenter le taux de perte.

Le deuxième type de réseau tout optique est les réseaux à commutation de paquet (Optical Packet Switching, OPS). Dans OPS [38], les données sont transportées dans des paquets optiques qui sont précédés par des en-têtes contenant les informations nécessaires pour le routage. Les nœuds intermédiaires s'occupent d'extraire les informations contenues dans l'entête du paquet et l'acheminer vers sa destination. Ce type de réseau se caractérise par un haut débit et s'adapte facilement à la variation du trafic. Cependant, le principal problème dans la commutation de

paquets est que le traitement des en-têtes se fait dans le domaine électrique ce qui nécessite le retardement des paquets optiques. Ceci nécessite donc l'utilisation des mémoires tampons optiques telles que les lignes de retard (FDL, Fibre Delay lines) qui sont coûteuses et exigent un grand espace.

Un troisième type de commutation a été introduit afin de surmonter les limites des deux commutations précédentes. La commutation optique par rafales (OBS) représente donc une solution prometteuse qui se situe entre OPS et OCS en combinant leurs avantages respectifs. Cette technique utilise efficacement la capacité des liens sans la nécessité de stocker les données ou d'effectuer la conversion optique-électronique-optique à chaque nœud intermédiaire. Ainsi, contrairement à OCS, OBS exige moins de complexité technologique que celle de la commutation optique par paquet et exploite mieux la bande passante.

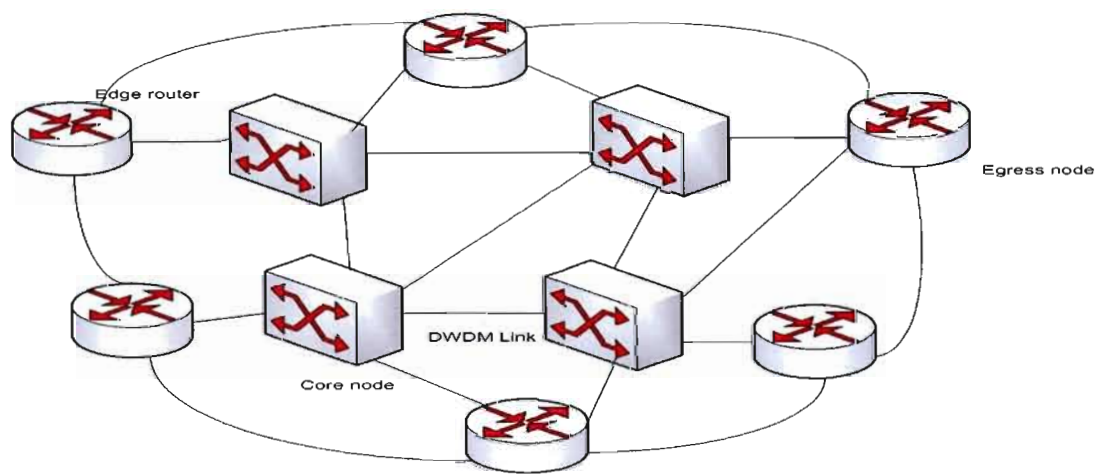
Dans ce qui suit, nous présenterons une description détaillée de l'architecture et les fonctions de la commutation optique par rafales.

### **2.1.1 Architecture générale d'OBS:**

OBS (*Optical Burst Switching*) est une technique de commutation de données dans les réseaux optiques (13). Elle représente une solution intermédiaire puisqu'elle permet de compenser les faiblesses des deux commutations: commutation par paquets (OPS) et commutation par circuit (OCS).

La commutation par rafales permet donc d'utiliser efficacement la bande passante qui est peu utilisée dans la commutation de circuits et aussi de diminuer le besoin des mémoires tampon (buffers) pour la commutation de paquets. Dans les réseaux OBS, les paquets ayant la même adresse de destination et quelques attributs communs (ex : conditions de qualités de service QoS) sont rassemblés dans une rafale (ou *Burst*). Chaque rafale a un paquet de contrôle qui est envoyé en premier pour la réservation des ressources et qui contient toutes les informations nécessaires pour son acheminement. Le temps entre l'envoi du paquet de contrôle et le burst est appelé *retard* (ou *Offset time*). Ce retard permet au paquet de contrôle de réserver les ressources nécessaires le long du chemin avant l'arrivée du burst. De cette façon, le burst transite dans le réseau optique sans conversion électronique. La séparation entre le paquet de contrôle et les données permet ainsi une meilleure gestion du réseau et plus de flexibilité.

L'architecture générale d'un réseau OBS se compose de routeurs électroniques (nœuds périphériques d'entrée ou sortie) et de routeurs optiques (nœuds internes) [14] reliés par des liens WDM (figure 1). Les fonctions fondamentales des nœuds périphériques sont le tri, la mise en mémoire et l'assemblage des paquets dans des bursts. Les nœuds internes se chargent de la signalisation, la résolution de la contention qui peut se produire quand plusieurs burst arrivent au même port de sortie en même temps. Les principales méthodes existantes pour résoudre la contention incluent la déflexion [28], [29], [30], la conversion de longueur d'onde [31], l'utilisation des FDL [32], [33] et la segmentation [34].



**Figure 2:** Architecture d'un réseau OBS

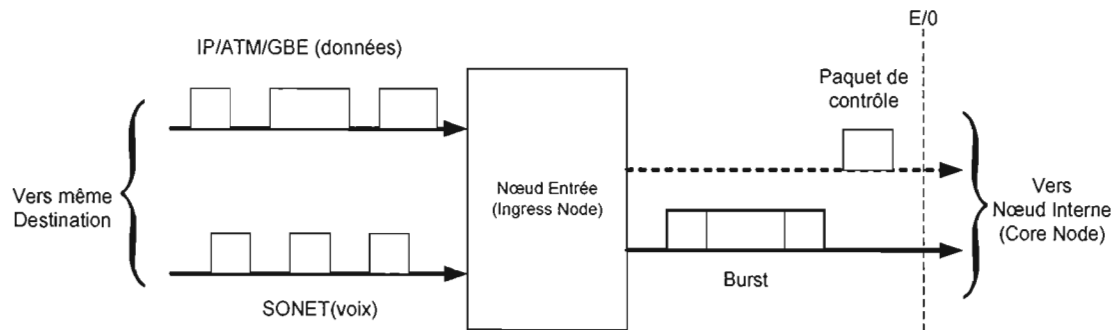
## 2.1.2 Concepts de base

### 2.1.2.1 Agrégation des bursts

Un des concepts importants dans les réseaux OBS est l'agrégation des bursts (*Burst assembly*). Une fois que le trafic des utilisateurs arrivent au nœud périphérique, ils sont d'abord triés selon leur classe de qualité du service et leurs adresses de destination puis assemblés dans des rafales et envoyés dans le réseau en tant qu'une seule entité. La Figure 2 illustre le mécanisme d'agrégation.

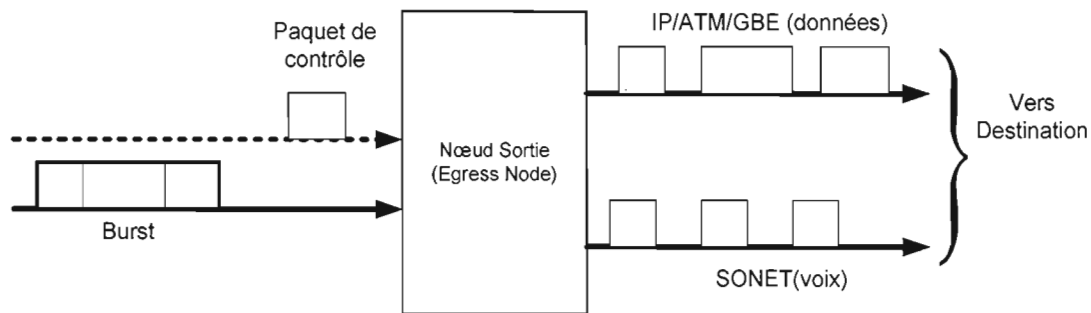
Il existe plusieurs méthodes d'agrégation basées sur différents paramètres qui sont : le temps d'agrégation et la taille de la rafale.





**Figure 3:** Agrégation des bursts (*Burst Assembly*)

Quand le burst de données arrive à son nœud de sortie, il est simplement décomposé en plusieurs paquets. Ensuite, chaque paquet est transmis à sa destination appropriée.



**Figure 4:** Burst Disassembly

#### ➤ Agrégation basée sur le temps

Cette méthode consiste à utiliser un temporisateur au début de chaque nouveau cycle d'agrégation et définit quand les paquets seront rassemblés en une rafale. Quand le temporisateur atteint zéro, la rafale est créée et acheminée vers sa destination. Plusieurs algorithmes d'agrégation basés sur le temps ont été développés. Par exemple, dans [1] les auteurs ont proposés trois algorithmes différents :

- L'algorithme FAP (*Fixed-Assembly-Period*) consiste à rassembler, dans un même burst, des paquets qui ont la même destination et qui arrivent pendant un intervalle de temps  $T$  fixe. Cependant, lorsque la charge du trafic est élevée, les bursts construits sont très larges.

- L'algorithme AAP (*Adaptive-Assembly-Period*) est similaire à l'algorithme précédent ; la seule différence est que le temps d'agrégation des bursts varie dynamiquement selon la longueur de la dernier burst reçu et ainsi il prend en compte l'intensité du trafic.
- L'algorithme MBMAP (*Min-BurstLength-Max-Assembly-Period*) utilise deux valeurs pour assembler les bursts. La première valeur est la taille minimale des bursts (*LMB*). La deuxième valeur représente le temps maximum d'assemblage d'un burst (*PMA*).

Le choix de la valeur du temporisateur est un critère très important dans le mécanisme d'agrégation. D'une part, si le temps est très petit cela va générer des petits bursts dans le réseau ce qui augmentera la contention. Et d'autre part, si la durée d'agrégation est grande, le taux de perte sera élevé. Ainsi, si les bursts sont assemblés dans un temps optimal, cela peut réduire les délais de bout en bout et améliorer la performance du réseau.

#### ➤ Agrégation basées sur la taille

Dans cette méthode un seuil (minimum ou maximum) est défini comme paramètre pour déterminer quand le burst sera construit et envoyé dans le réseau. Le seuil spécifie le nombre de paquets à rassembler dans un burst. Les paquets sont stockés dans des files d'attente dans le nœud d'entrée jusqu'à ce que le seuil soit atteint [2][3]. Si la longueur du burst est longue, cela va allonger la durée de création et mener à plus de perte de données lors d'une perte de burst. Par contre, si longueur du burst est petite cela peut cause une surcharge du réseau puisque l'interarrivée entre les paquets de contrôle sera très petit. Il est donc important de bien définir le seuil d'agrégation afin d'optimiser la performance de tout réseau d'OBS avec des paramètres réseau spécifiques.

Dans [3], une nouvelle technique d'agrégation basée sur la taille a été proposée. Elle consiste à assembler dans un même burst tous les trafics utilisateurs ayant les mêmes exigences QoS. Les simulations ont été exécutées en variant les valeurs du seuil et priorité des burst. Il était constaté qu'il y a un seuil optimal qui minimise la probabilité de perte pour un réseau spécifique et avec une charge donnée.

### 2.1.2.2 Routage et assignement de longueur d'onde (RWA)

Au sein d'un réseau OBS, la décision du routage des bursts est prise à chaque nœud en calculant le plus court chemin vers chaque destination. Trois méthodes de routages sont proposées :

- La méthode est « hop-by-hop » [17], comme dans un réseau IP, il consiste à exécuter un algorithme de consultation rapide pour déterminer le prochain saut du burst.
- Le protocole MPLS (*Multi-protocol label switching*) proposée par l'IETF [18]. Dans le routage MPLS basé OBS, chaque paquet de contrôle est assigné à sa classe de trafic (FEC : Forwarding Equivalence Class) au nœud source. Ensuite, les nœuds intermédiaires acheminent le burst vers son étiquette.
- Le routage basé sur l'utilisation des chemins pré-calculés obtenus par le protocole RSVP (protocole de réservation de ressource) ou CR-LDP (protocole Contrainte-Basé de distribution d'étiquette d'itinéraire) [19].

L'assignement de la longueur d'onde est une des techniques utilisées pour déterminer à quelle longueur d'onde en particulier transmettre le burst. Cet assignement peut être fait avec ou sans conversion. La première approche [16] est d'assigner la même longueur d'onde le long du chemin entier de la source à la destination. Cependant, l'inconvénient principal de cette méthode est un taux de perte élevé dû à l'indisponibilité de la longueur d'onde choisie par le chemin entier. Dans la deuxième approche [15] avec conversion de longueur d'onde, chaque nœud OBS peut assigner une longueur d'onde sortante différente à chaque arrivée du burst en se basant sur l'ensemble de longueurs d'onde disponibles. Dans ce cas, les nœuds d'OBS transmettent les signaux optiquement d'une longueur d'onde entrante à une longueur d'onde sortante différente ce qui réalise une baisse importante du taux de perte dans le réseau.

### 2.1.3 Signalisation

La signalisation représente un concept important dans la commutation par rafale. Ce mécanisme permet d'allouer des ressources et de configurer les routeurs optiques à chaque nœud. Plusieurs algorithmes de signalisation ont été proposés dans la littérature tels que : TAG (Tell-And-Go), JET (Just-Enough-Time), JIT (Just-In-Time), Horizon et WR-OBS.

### 2.1.3.1 TAG (*Tell-And-Go*)

Dans le mécanisme *TAG* [6], le burst de données est envoyé immédiatement après le paquet de contrôle (contenant les adresses source et destination) sans attendre la réception d'un acquittement. A chaque routeur, des FDLs sont employées pour retarder le burst pendant que le paquet de contrôle réserve la longueur d'onde et ensuite le transmettre sur la longueur d'onde réservée. L'avantage du TAG est que la transmission des données se fait sans attente de confirmation. Dans un réseau à grande vitesse, où le temps de traitement du paquet de contrôle et le temps de configuration des routeurs sont négligeables, cette technique est efficace. Mais son inconvénient est que la probabilité de blocage est élevée puisque le burst peut être rejeté avant que les ressources ne soient réservées par le paquet de contrôle.

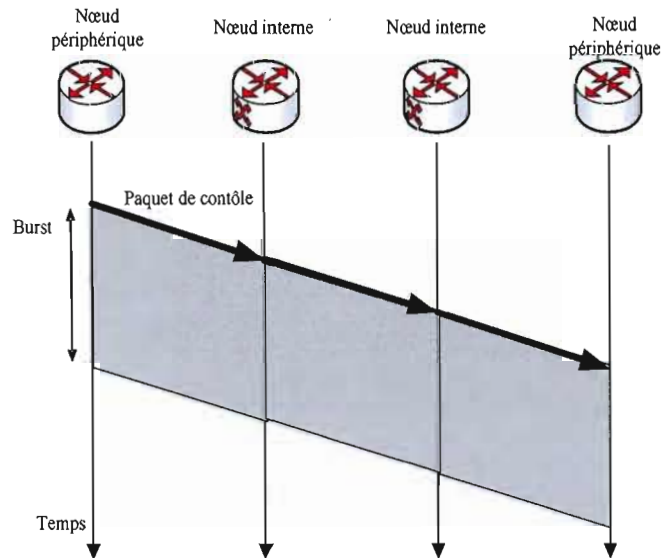
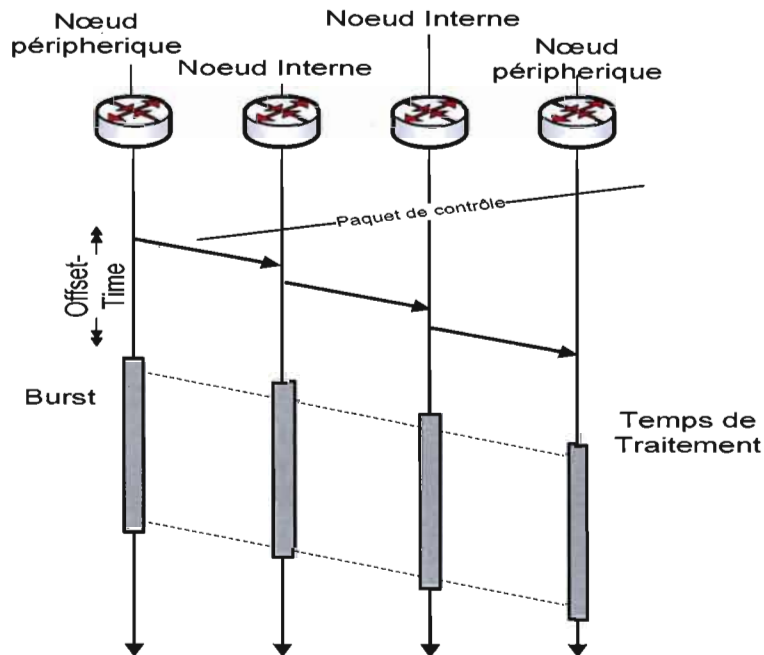


Figure 4: Tell and Go (TAG)

### 2.1.3.2 JET (*Just-Enough-Time*)

JET est basé sur la réservation retardée et la libération implicite et n'exige aucune mise en mémoire ou délai aux nœuds intermédiaires [7]. Le paquet de contrôle, contenant la taille du burst, réserve la bande passante au moment de l'arrivée du burst jusqu'à la fin de sa transmission, et donc les ressources sont libérées directement après la transmission du dernier bit du burst.

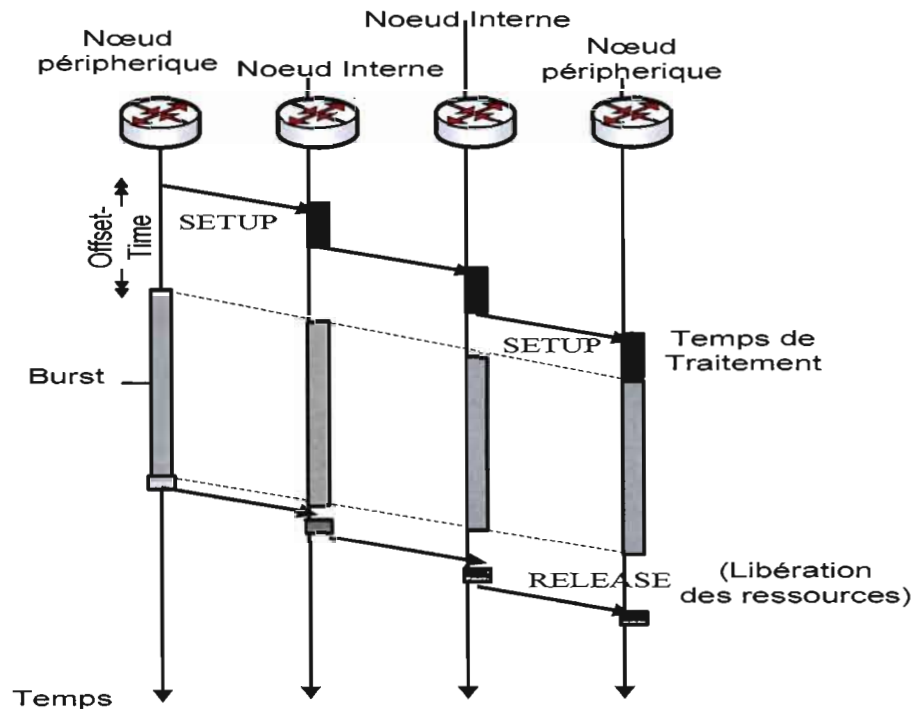
Contrairement au TAG, la transmission du burst est retardée à la source (dans le domaine électronique) durant une durée définie « offset-time » ce qui peut éliminer le besoin d'utiliser des FDLS ou optimiser leur utilisation. Ce qui augmente l'utilisation de la bande passante et réduit les retransmissions.



**Figure 5:** Just-Enough-Time (JET)

### 2.1.3.3 JIT (*Just-In-Time*)

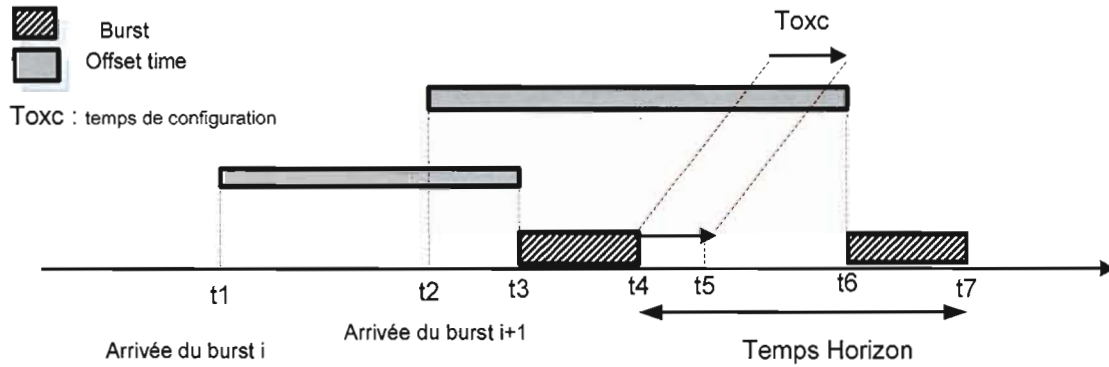
JIT est une technique de signalisation unidirectionnelle qui consiste en la réservation immédiate et la libération explicite(10). La longueur d'onde est réservée pour un burst juste après l'arrivée du message d'établissement de connexion correspondant (SETUP) ; si une longueur d'onde ne peut pas être réservée, alors le message SETUP est rejeté et le burst correspondant est détruit. L'avantage principal du JIT est qu'il minimise le délai bout-à-bout pour la transmission de données et donne une meilleure performance. Cependant il peut mener à une probabilité de blocage dû à l'allocation inefficace de la bande passante.



**Figure 6:** Just-In-Time (JIT)

#### 2.1.3.4 Horizon

Le protocole de signalisation « horizon » utilise la réservation retardée [9]. Chaque longueur d'onde est associée à un temps Horizon pour la réservation des ressources. Quand un burst est cédulé pour une longueur d'onde donnée, le temps Horizon est mis à jour au moment du départ du burst plus le temps configuration du routeur optique OXC. Par conséquent, un nouveau burst ne sera programmé sur une longueur d'onde que si et seulement si le premier bit du burst arrive après tous les burst actuellement programmés sur cette longueur d'onde seront partis. Dans le cas ou aucune longueur d'onde n'a un temps horizon plus petit que son temps d'arrivée, le burst est donc rejeté [20]. L'algorithme d'horizon est relativement simple et réalise de bons résultats en termes du temps d'exécution [21]. Cependant, il en résulte une mauvaise utilisation de la bande passante et un taux de perte élevé. Ceci est dû au fait qu'horizon rejette tous les intervalles vides.

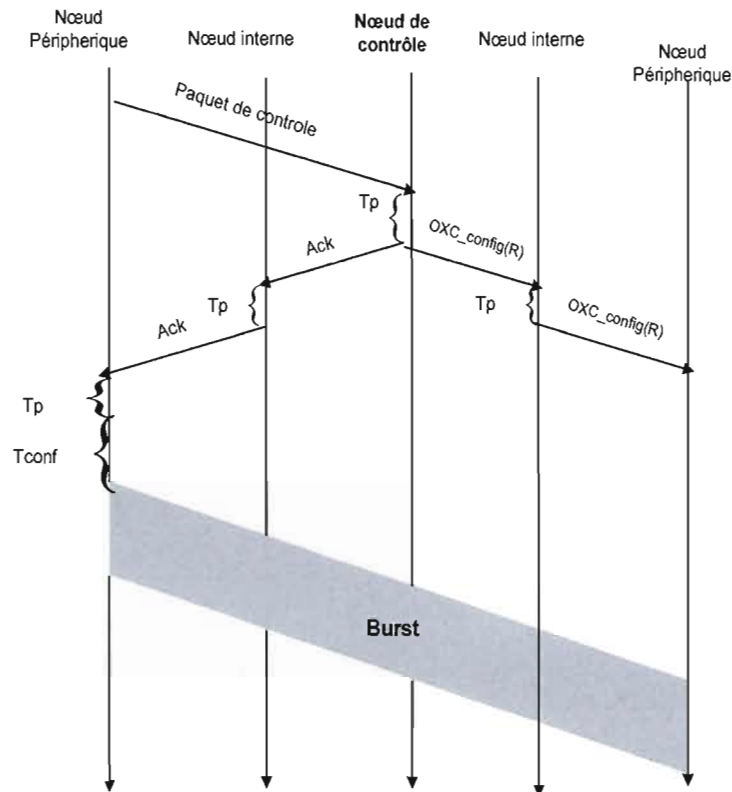


**Figure 7:** Horizon

La figure 6 illustre la transmission de deux bursts successivement sur la même longueur d'onde en utilisant Horizon. Le message SETUP du burst  $i$  arrive au nœud OBS à l'instant  $t_1$  et le dernier bit de ce burst quitte le nœud à l'instant  $t_4$ .  $Toxc$  représente le temps de configuration du routeur optique. Aucun burst ne peut être transmis sur cette longueur d'onde pendant un temps  $t_5 = t_4 + Toxc$ . Ainsi, le temps Horizon est initialisé à  $t_5$  dès la réception du premier bit du burst  $i$ . Quand le message Setup du burst  $i+1$  arrive au nœud à l'instant  $t_2 > t_1$ . Le nœud utilise l'information contenue dans le message SETUP pour calculer le temps d'arrivée du dernier bit de ce burst  $t_6$ . Si  $t_6 > t_5$  le burst sera transmis sur la même longueur d'onde et le temps Horizon est mis à jour et sera égale à  $t_7 + Toxc$  ( $t_7$  est le temps d'arrivée du dernier bit du burst  $i+1$ ).

### 2.1.3.5 WR-OBS

Contrairement aux protocoles de signalisation unidirectionnelle, une méthode de signalisation centralisée a été proposée dans [8], appelée « Commutation de bursts optiques intégrés au routage de longueurs d'onde » (WR-OBS). Cette méthode utilise une procédure de réservation des ressources de bout en bout. Les paquets de contrôle sont envoyés et traités à un nœud central pour déterminer les chemins appropriés et assigner des longueurs d'onde disponibles le long de chaque lien. Une fois le paquet est traité, un acquittement est envoyé à la source. A partir de cette réception, le nœud source transmet le burst correspondant. Les principaux avantages de cette architecture sont l'approvisionnement explicite de QoS, l'adaptation rapide aux changements dynamiques du trafic dans les réseaux optiques et aussi l'utilisation efficace de chaque canal de longueur d'onde [22].



**Figure 8: WR-OBS**

## 2.2 Grilles de calcul :

### 2.2.1 Introduction

Avec la grande évolution des technologies, il est devenu évident que les ressources informatiques ne peuvent pas répondre à tous les besoins des utilisateurs et des applications. La plupart des applications scientifiques et économiques ont des grandes exigences informatiques. De là, le terme « Grille » a été introduit à la fin des années 90 pour désigner une infrastructure répartie proposée pour la science avancée et l'ingénierie dans [23]. Les grilles visent à résoudre ce problème en fournissant une infrastructure flexible (infrastructure matérielle et de logicielle) et pratique qui offre un accès sûr, peu coûteux, performant et sécurisé aux ressources. Elles se composent de ressources hétérogènes géographiquement distribuées (informatiques, stockage,



données, etc.), interconnectés ensemble par différents types de réseaux et offre une interface uniforme à ces ressources [24]. Le fonctionnement de la grille est donc semblable au concept du Web. Le Web est un service pour partager l'information via Internet tandis que la grille est un service pour partager des ressources de stockage ou des données via Internet. L'avantage des grilles consiste en un traitement plus rapide et plus efficace des applications comportant de nombreux calculs et nécessitant de grandes ressources de stockage. Elles permettent aussi d'éliminer l'inconvénient de lier étroitement des machines spécifiques à des travaux spécifiques. Puisqu'elles permettent à des utilisateurs et à des applications d'accéder et contrôler efficacement diverses ressources de différentes locations distribuées.

Les domaines d'application des grilles sont essentiellement divisés en cinq grandes classes [25].

- Le calcul intensif distribué : Additionner les puissances de calcul non utilisées de plusieurs supercalculateurs ou simplement des stations de travail afin de résoudre un problème donné (par exemple : simulation).
- Le calcul à haut débit : Ordonnancer un grand nombre de tâches, peu ou pas du tout couplées, sur des machines inutilisées pour la résolution d'un problème unique ou des problèmes différents.
- Le calcul à la demande : Utiliser temporairement de ressources (exp : capacités de calcul, logiciels, bases de données...).
- Le traitement intensif de données : Produire de nouvelles informations à partir de données géographiquement distribuées comme dans le cas des systèmes de prévisions météorologiques, des programmes d'observation terrestre...etc.
- Le calcul collaboratif : Mettre en place des interactions entre personnes humaines pour l'exploration simultanée de bases de données, des systèmes de réalité virtuelle à objectifs éducatifs ou de distraction.

Aujourd'hui il existe de nombreuses contributions dans le domaine des grilles informatiques dans différents domaines de recherches, tels que Physique nucléaire (Datagrid [26], Eurogrid [27]), Bioinformatique (Datagrid, Eurogrid), Météorologie (Eurogrid), Sciences de l'univers et observation de la terre, Santé (imagerie médicale, cancer...), Sciences (écologie, risques financiers...), etc.

### 2.2.2 Architecture générale de la grille

L'architecture proposée dans [23] est d'un niveau d'abstraction élevé et n'exige pas beaucoup de contraintes en termes de design et implémentation. Elle identifie les composants systèmes fondamentaux, leurs fonctions et montre comment ces composants interagissent entre eux. Dans ce qui suit nous énumérons les différentes couches de la grille et dériverons brièvement le rôle de chacune.

La figure 9 décrit les couches de la grille et expose la similitude existante entre l'architecture de la grille et le modèle OSI puisque qu'elle s'étend de la couche réseau à la couche application.

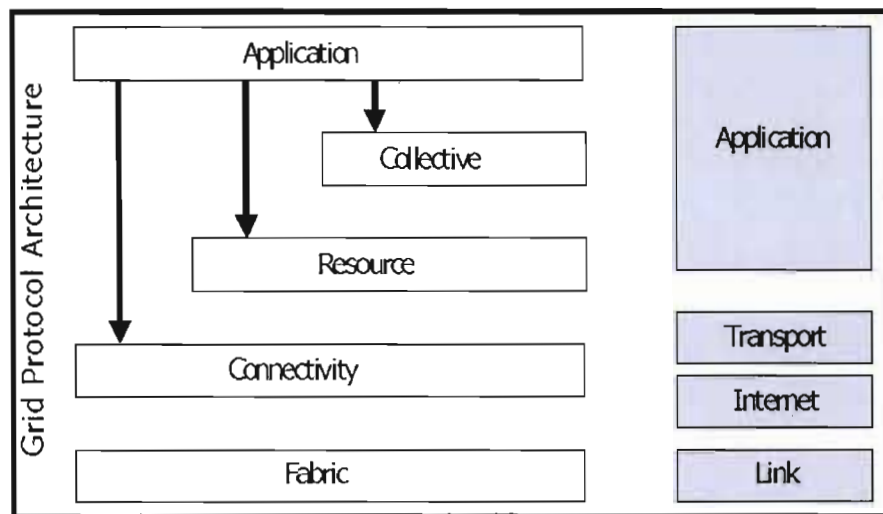


Figure 9: Architecture de la grille et modèle OSI [23]

#### Fabrique

La couche « *fabrique* » comprend les ressources de calcul (clusters, machines de bureau, etc.), les systèmes de stockage, les réseaux de transmission qui vont être partagés via les protocoles de la grille. Une ressource peut être aussi une entité logique, telle qu'un système de fichiers distribué, ce qui peut impliquer l'utilisation d'autres protocoles.

Les composants de la couche Fabrique implémentent les opérations pour tout accès aux ressources. Il existe ainsi une dépendance entre les fonctions implémentées au niveau du Fabrique et les opérations de partage. Par conséquent, une fonctionnalité plus développée de cette couche

permet des opérations de partage plus sophistiquées ; en même temps, s'il y a peu de demandes à traiter, le déploiement de l'infrastructure de grille sera plus simple.

## **Connectivité**

La couche Connectivité définit les protocoles de communication et d'authentification utilisés dans les transactions à l'intérieur de la grille. Les protocoles de communications permettent l'échange des données entre la couche fabrique et la couche ressource. Les protocoles d'authentification, liés aux services de communication, fournissent les mécanismes de sécurité pour vérifier l'identité des utilisateurs et des ressources.

Dans les grilles, la sécurité est un aspect plus important que dans les environnements informatiques traditionnels à cause du grand nombre de ressources partagés et qui sont de plus réparties sur divers sites géographiques. Plusieurs solutions de sécurité et communication de la grille sont basées sur des normes existantes (la suite des protocoles TCP/IP).

## **Resource**

La couche Ressource se base sur les protocoles de communication et d'authentification pour définir des protocoles pour la négociation sécurisée, le déclenchement, le contrôle, la surveillance, le compte, et le paiement des opérations de partage sur différentes ressources. Les protocoles de la couche ressource appellent les fonctions de la couche de Fabrique pour accéder et contrôler les ressources locales. Ces protocoles prennent en charge toute ressource individuelle et ignorent tout traitement de l'état global des ressources. On distingue deux principales classes des protocoles de la couche de ressource : les protocoles de l'information et de gestion. Les protocoles de l'information sont utilisés pour obtenir des informations sur la structure et l'état d'une ressource, par exemple : sa configuration, charge actuelle, et politique d'utilisation (coût). Les protocoles de gestion sont employés pour négocier l'accès à une ressource partagée, par exemple, les exigences de la ressource (réservation avancée et qualité du service) et les opérations à exécuter (lancement de processus ou l'accès aux données). Ils doivent s'assurer aussi que la politique de partage de la ressource est respectée.

## **Collective**

Tandis que la couche de ressource est concentrée sur des interactions avec une ressource individuelle, la couche collective comprend les protocoles et les services qui sont associés aux interactions entre une collection de ressources. Elle offre ainsi une variété de services sans exiger des nouvelles conditions pour le partage de ressources. Par exemple :

- découvrir l'existence et les propriétés des différentes ressources d'une organisation virtuelle (service d'information et d'annuaire) ;
- faire de la co-allocation et de l'ordonnancement, c'est-à-dire permettre aux clients de la grille de demander l'allocation d'une tâche particulière à une ou plusieurs ressources, selon une certaine logique ;
- transférer les données nécessaires aux tâches d'un site vers un autre ;
- gérer la facturation des services.

## **Applications**

La couche située au niveau le plus élevé de la structure de la grille est la couche application et qui comprend différentes applications utilisateurs (scientifiques, commerciales, de gestion...etc.). Elle contient aussi les outils nécessaires afin d'aider les développeurs à concevoir et à écrire des applications adaptées à la grille (compilateurs, bibliothèques, interfaces de programmation...etc.).

## CHAPITRE III

### GRILLE BASEE SUR OBS (GOBS)

#### 3.1 Introduction

L'introduction des grilles informatiques a permis la résolution des problèmes de capacité de calcul et d'espace de stockage. Cependant, le principal défi des réseaux grilles actuels et les futures grilles est le choix de l'infrastructure technologique qui dépend des caractéristiques des applications. Les applications de la grille peuvent différer selon plusieurs caractéristiques : granularité du trafic, la bande passante, QoS, le délai, le taux de perte, la capacité de traitement, etc. Les réseaux optiques peuvent représenter une solution idéale pour répondre aux demandes des différents types d'applications. Toutefois, la fusion entre les technologies optiques et les grilles doit fournir les services suivants [39] :

- Une infrastructure réseau flexible, reconfigurable et évolutive.
- Une grande capacité de traitement : Transfert de données en bloc.
- Une bande passante disponible à chaque demande pendant des périodes courtes ou longues et entre différents emplacements dans le réseau.
- Un débit variable et constant.
- Un approvisionnement des longueurs d'onde et portions d'onde au niveau des ressources.
- Des mécanismes du Broadcasting/multicasting/anycasting : afin d'assigner les tâches de la grille aux multiples ressources de la grille.
- Une plateforme matérielle flexible pour supporter différents types des ressources (stockage, calcul, etc.).
- Une sécurité aux deux niveaux : réseau et utilisateur.

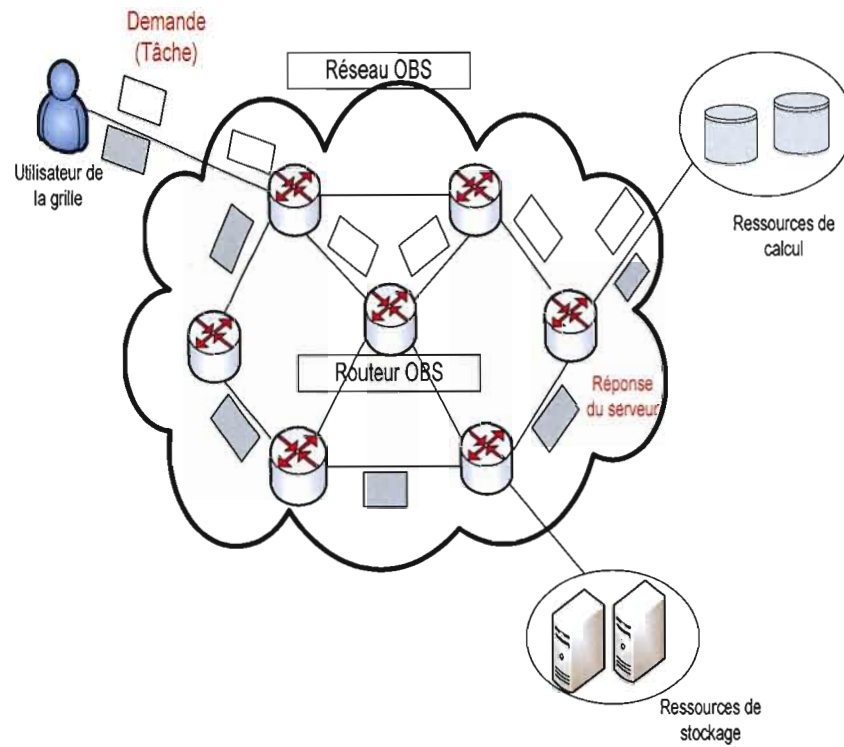
Un protocole de transport approprié aux grilles optiques et qui répond aux exigences citées ci-dessus est la commutation optique par burst (OBS) [39]. OBS peut être employé pour supporter différents types de trafic multiservices ce qui offre une granularité et efficacité élevées. Il fournit la convergence des technologies électroniques et optiques; ce qui permet une infrastructure réseau apte d'assurer les fonctionnalités de la grille (tels que : la découverte des ressources, la sécurité etc). De plus, OBS offre la possibilité de traiter et s'adapter à différents types de tâches (petites ou grandes) puisque la taille du burst peut varier selon les exigences des applications en termes de durée, latence, etc. Une autre caractéristique importante d'OBS est qu'il n'impose pas de condition stricte pour la symétrie dans les deux directions de la transmission (utilisateur  $\Leftrightarrow$  Ressource).

Dés la première proposition initiale [40], les grilles basées OBS ont attirées l'attention d'une large communauté de chercheurs et différents travaux et solutions ont été proposées concernant l'architecture appropriée [39] [40] , les mécanismes de signalisations [44], les algorithmes de routage [35] etc.

### **3.2 Architecture d'un réseau GOBS :**

Dans cette section, nous présentons l'architecture d'un réseau grille utilisant la technologie d'OBS [39] [40] où les nœuds OBS exécutent la découverte des ressources et l'acheminement des tâches de la grille. Ces Tâches représentent les demandes des utilisateurs indiquant toutes les exigences requises pour le traitement de leurs applications. Ces tâches sont transformées en un burst et qui est précédé par un paquet de contrôle, qui contient les caractéristiques de la tâche énuméré dans le tableau1. Les tâches sont traitées initialement au nœud d'entrée et donc les fonctions de découverte et d'allocation des ressources doivent être exécutées. Dès qu'une tâche est reçue par le nœud d'entrée, il vérifie les ressources disponibles qui répondent aux exigences de la tâche et décide si la tâche sera envoyée dans le réseau ou non. Les nœuds intermédiaires ne sont pas alors avertis d'avance de l'arrivée d'une tâche (c.-à-d. aucune réservation à l'avance), mais décident au fur à mesure où expédier les données. Il est important de noter qu'une tâche en générale n'est pas traitée dans un seul endroit, le réseau doit pouvoir allouer toutes les ressources disponibles dans différents emplacements afin de satisfaire les exigences de toutes les applications de stockage et/ou de traitement. Cependant, l'implémentation d'un réseau GOBS implique de nouvelles contraintes pour une transmission intelligente et rapide. Ainsi, un certain nombre de caractéristiques additionnelles s'ajoutent aux routeurs réguliers d'OBS. Dans ce qui

suit nous présenterons donc l'infrastructure détaillée d'un réseau GOBS et le rôle de chaque composant.



**Figure 10:** Architecture de d'un réseau GOBS.

Champ	Description
type	Le type du burst peut être soit une réponse ou une demande (une tâche traité ou non)
ID	L'identification du burst utilisé pour la séquence des Tâches
Canal d'entrée (Ingress Channel)	La longueur d'onde utilisée par le burst
Durée	La durée en temps du burst
Offset	Offset entre le burst et son paquet de contrôle
Destination	L'adresse destination du nœud intermédiaire
Historique du Routage	Le nombre de sauts, le chemin pris par le burst, etc
Durée de traitement	Le temps exigé pour le traitement d'une Tâche ou global du traitement
Délai d'expiration	Le délai (end-to-end) maximum avant que le burst expire

**Tableau 1 :** Description des champs du paquet de contrôle [49]

### 3.2.1 Nœud périphérique

Un nœud périphérique dans un réseau GOBS doit répondre aux exigences des applications de la grille et aussi offrir une utilisation efficace des ressources réseau en employant toutes les



fonctionnalités d'OBS. Ainsi, les principales tâches qui s'ajoutent aux fonctions d'un nœud périphérique dans GOBS [43] sont les suivantes:

- Classification des tâches de la grille.
- Coordination entre les ressources réseau et les utilisateurs de la grille.

### **3.2.1.1 Classification des tâches de la grille**

La première fonction exécutée par un nœud d'entrée est la classification des tâches. Le principe de classification est de différencier les besoins d'un utilisateur/d'une application de la grille afin de fournir des services adéquats et spécifiques. Puisqu'en assignant les tâches utilisateurs aux bonnes ressources réseau et informatiques qui répondent à leurs exigences, peut facilement améliorer la performance au sein de la grille. Il est donc important de définir un mécanisme de classification flexible et extensible. La classification des tâches est divisée en trois types : classification orientée réseau (*Network-Oriented classification*), Classification Orientée Grille (*Grid-Oriented Classification*) et une Classification basée sur le temps (*Time-Oriented Classification*). Ces trois opérations sont traitées parallèlement et indépendamment dès qu'une demande grille est reçue au nœud périphérique.

Le mécanisme de classification au sein de la grille vise à faciliter la découverte et l'allocation des ressources. Ce qui implique l'introduction de mécanismes intelligents aux routeurs réguliers d'OBS pour assurer l'ordonnancement et la découverte des ressources. Toutefois, cette classification doit répondre à d'autres exigences telles que la vitesse, le taux de mise à jour etc. Par exemple, si la durée de classification est longue ceci aura comme conséquence la mise en file d'attente des paquets de contrôle avant qu'ils soient traités ce qui augmente le taux de perte [45].

### **3.2.1.2 Coordination entre les utilisateurs/ ressources de la grille et le réseau**

Pour faciliter l'accès aux services de la grille, deux interfaces sont intégrées dans l'architecture du réseau GOBS. La première est l'interface réseau pour les utilisateurs de la grille (GUNI) qui représentent des procédures interopérables entre les utilisateurs et le réseau optique pour la négociation et l'activation d'un service. La deuxième est l'interface réseau pour les ressources de grille (GRNI) qui assure l'accès aux ressources de calcul ou de stockage et aussi la diffusion de

disponibilité des ressources. La figure 11 représente l'infrastructure d'un réseau GOBS qui incorpore les deux interfaces GUNI [39] et GRNI [44].

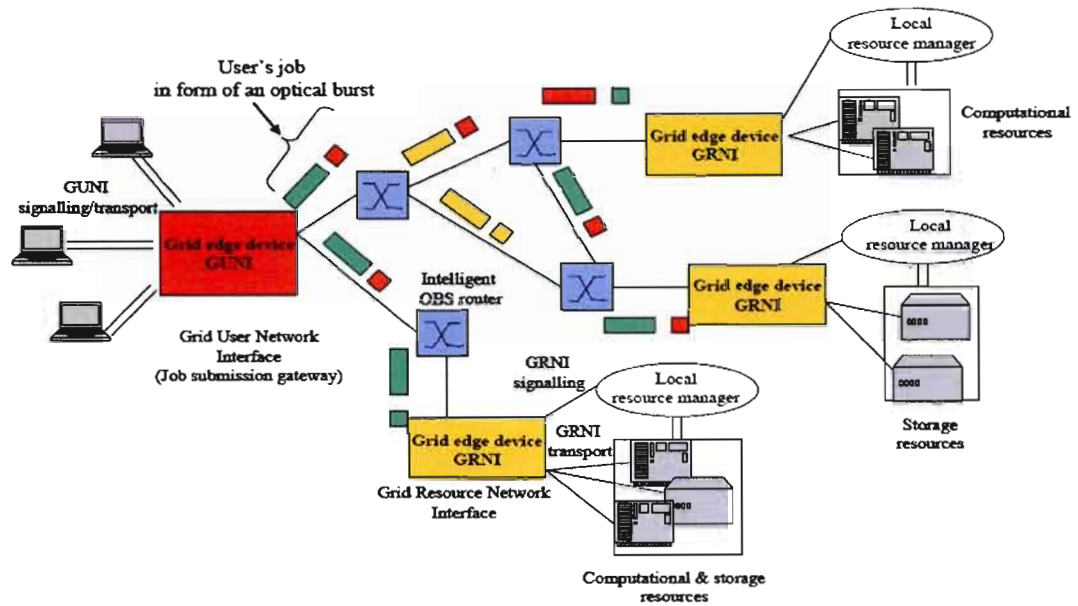


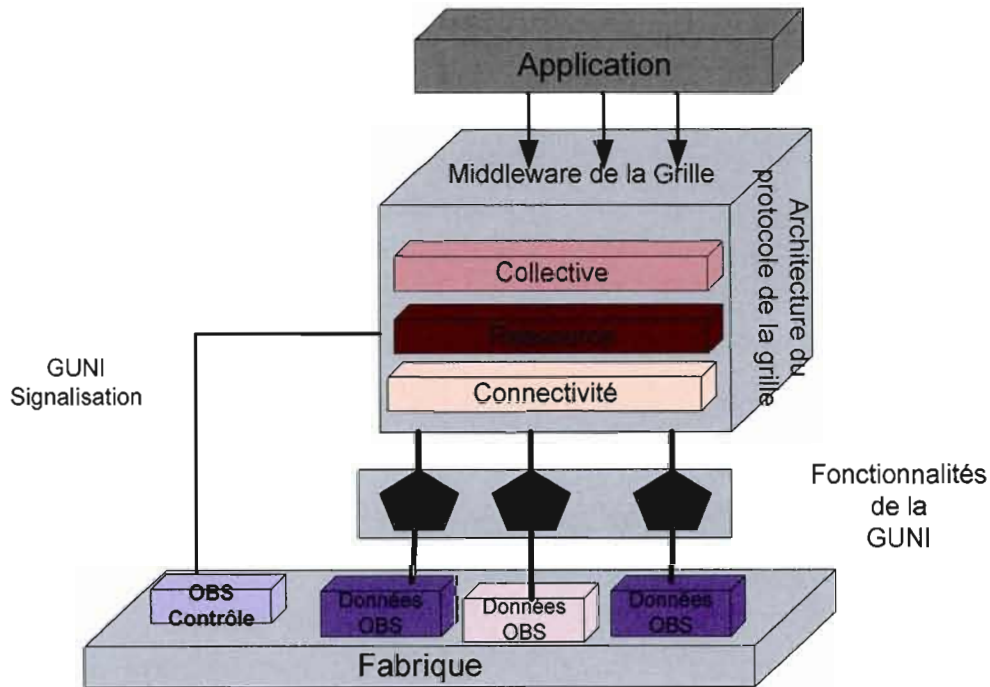
Figure 11: Infrastructure d'un réseau GOBS [40]

#### Interface réseau pour les utilisateurs de la grille (GUNI):

GUNI sera une partie fondamentale du nœud périphérique afin de faciliter l'accès des utilisateurs au réseau optique et assurer aussi des services intelligents. Les fonctionnalités principales fournies par GUNI [39] sont les suivantes :

- Allocation flexible de la bande passante : Elle fournit un mécanisme pour l'allocation de la bande passante requise (c.-à-d. Longueur d'onde ou sous-longueur d'onde) pour un utilisateur de la grille ou un service.
- Établissement automatique du chemin : Les utilisateurs peuvent automatiquement céder, choisir, et établir le chemin de transmission à travers (lightpath) le réseau.
- Agrégation et transmission des bursts : La GUNI exécute la classification et l'agrégation du trafic en une entité de transmission (burst).

- Découverte et ordonnancement des ressources automatiques: La GUNI exécute une méthode de découverte et d'ordonnancement des ressources en conjonction avec l'architecture du protocole de la grille en employant le paquet de contrôle (c.-à-d. signalisation de longueur d'onde).
- Sécurité : La GUNI offre un mécanisme de sécurité au niveau de la signalisation et la transmission.



**Figure 12:** Interface réseau pour les utilisateurs de la grille (GUNI)

### **Interface réseau pour les ressources de la grille (GRNI):**

L'interface GRNI représente l'ensemble des procédures qui assurent toute communication entre les ressources et le réseau OBS. Elle fournit le mécanisme de signalisation entre les ressources locales et le réseau optique. Elle offre également un mécanisme de transport des données entre les ressources et le réseau. Le nœud périphérique reçoit donc une tâche sous forme de burst du réseau; ensuite, l'envoi dans le format approprié aux ressources et il exécute l'opération inverse dans le cas d'une tâche traitée. Ainsi les principales fonctionnalités de la GRNI sont les suivantes [40]. :

- Ségrégation des bursts entrants.
- Soumission de la tâche aux ressources de la grille.
- Propagation de l'état des ressources locales au réseau optique.
- Renvoi des résultats de la tâche traitée.
- Construction d'entité de transmission (Burst).
- Allocation de la bande passante et établissement du lightpath.

### 3.2.2 Les routeurs intermédiaires

Les routeurs intermédiaires (*core nodes*) représentent des éléments actifs au sein d'un réseau GOBS qui traitent le paquet de contrôle et offre une transmissions intelligente des données. Ils assurent ainsi le routage des bursts vers les nœuds appropriés et résolvent les problèmes de contention qui peuvent surgir. Ces opérations nécessitent donc l'utilisation de circuits électroniques conçus particulièrement pour l'exécution d'une commutation rapide qui pourra supporter les exigences des applications de la grille en termes de temps. Une des solutions proposées [43] [46] est un commutateur optique (Optical Cross Connect : OXC) qui combine des éléments de commutation lents et rapides et répondent aux exigences en termes de mise a l'échelle (*scalability*) et de coût. Ainsi, dès qu'un paquet de contrôle est envoyé dans le réseau, on identifie la taille et le type du burst afin de l'acheminer vers le port approprié (lent ou rapide). En vérifiant les ports disponibles, l'OXC est configuré pour recevoir le burst de données. Cependant, dans le cas où aucun port n'est disponible, le burst sera détruit.

### 3.3 Signalisation :

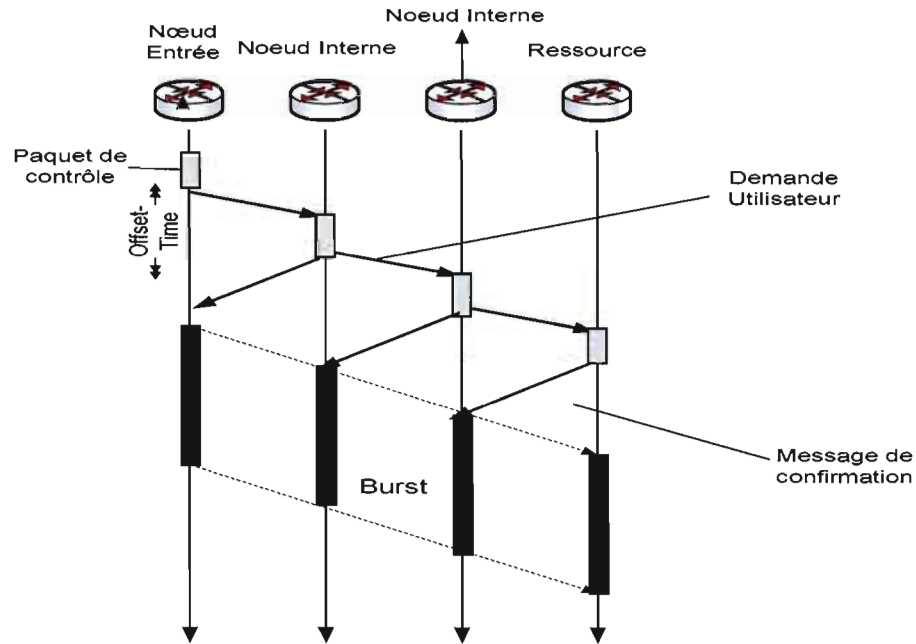
La signalisation consiste en le transport des messages de contrôle entre toutes les différentes entités du réseau. Au sein d'un réseau GOBS, la signalisation est utilisée pour la découverte et la réservation des ressources. Deux approches de réservation des ressources sont proposées [44] : unidirectionnelle ou bidirectionnelle.

Dans la réservation à sens unique (unidirectionnelle) une seule paire (paquet de contrôle, Burst) est envoyée pour découvrir et réserver les ressources. Le paquet de contrôle est envoyé en premier contenant la description des exigences de la tâche. Ce qui permet aux nœuds intermédiaires de sélectionner les ressources appropriées pour le traitement de la tâche et réserver la bande passante le long du chemin du client à la ressource. Le burst de données est envoyé donc peu de temps après le paquet de contrôle.

Cependant, dans la découverte à deux sens, un premier paquet de contrôle est envoyé pour la découverte et la réservation des ressources. Quand une ressource répondant aux exigences de la tâche est atteinte, un acquittement (ACK) est envoyé au client pour la transmission du burst de données. Dans le cas où aucune ressource n'est disponible un acquittement négatif (NACK) est envoyé.

### **3. 3.1 JET (Just Enough Time):**

Parmi les différents protocoles de signalisation, JET est le plus approprié pour les réseaux GOBS [50]. JET est une approche basée sur la réservation à un seul sens et qui utilise la réservation retardée. Ainsi une longueur d'onde est réservée pour un burst juste avant l'arrivée du premier bit. Toutefois, si aucune longueur d'onde n'a pu être réservée, le paquet de contrôle est rejeté et le burst correspondant est détruit. Dans [36], un schéma simple basé JET pour la découverte et la réservation des ressources a été proposé (Figure13). Dans ce schéma le paquet de contrôle est envoyé en premier pour réserver les ressources à chaque nœud intermédiaire et qui par la suite envoie un message de confirmation au nœud d'entrée. Une fois le message de confirmation est reçu, le burst de données est envoyé.



**Figure 13: JET-GOBS**

Dans d'autres travaux [50] [51] l'utilisation du JET a été modifiée afin de l'adapter au trafic de la grille en le transmettant en deux étapes ce qui assure une transmission efficace. Premièrement les spécifications de la tâche sont envoyées sous forme d'un burst actif ensuite la tâche utilisateur (données) est transmise sous forme d'un burst inactif. Ceci fournit une efficacité au niveau de la bande passante particulièrement quand un grand nombre de données doit être transféré vu que la tâche est envoyée juste quand les ressources sont réservées. Cette infrastructure sera détaillée ultérieurement dans le prochain chapitre.

### 3.3.2 JIT (Just In Time):

Une nouvelle architecture de GOBS qui réunit l'utilisation de commutateur optique et les technologies agiles avec des processeurs réseau (*Network processeur : NP*) à grande vitesse a été introduite dans [52]. Cette architecture utilise JIT comme protocole de signalisation pour la découverte et la réservation des ressources.

La Figure 15 illustre le schéma de signalisation proposé. L'utilisateur avec des données à traiter, envoie une demande au routeur périphérique en décrivant les spécifications de la tâche et les exigences du traitement (informatique et réseau). Le routeur d'entrée classe les demandes entrantes et puis construit et transmet la demande optique (tâche) au routeur intermédiaire. Dès sa réception, la demande est traitée par le NP installé dans le routeur intermédiaire. Il exécute donc un algorithme de découverte des ressources afin d'identifier les ressources disponibles. Ensuite, l'utilisateur est informé du résultat de la découverte de la ressource par un acquittement (ACK) ou un non-acquittement dans le cas où de ressources non disponibles ou insuffisantes. En envoyant un ACK, une réservation des ressources simultanément est envoyée au nœud destination. Ainsi l'utilisateur envoie les données au nœud d'entrée qui les rassemblent dans un burst optique et l'achemine vers les ressources réservées selon le fonctionnement du JIT.

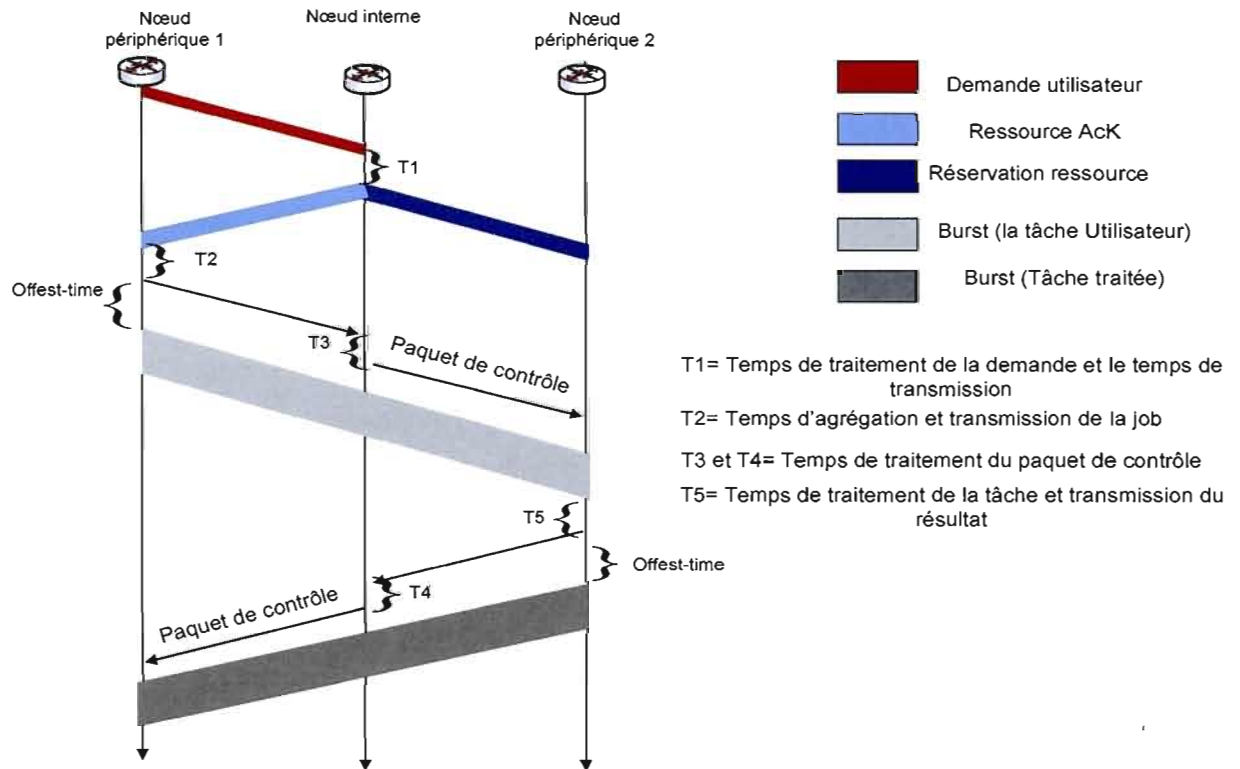


Figure 14: JIT-GOBS

### 3.3.3 Horizon

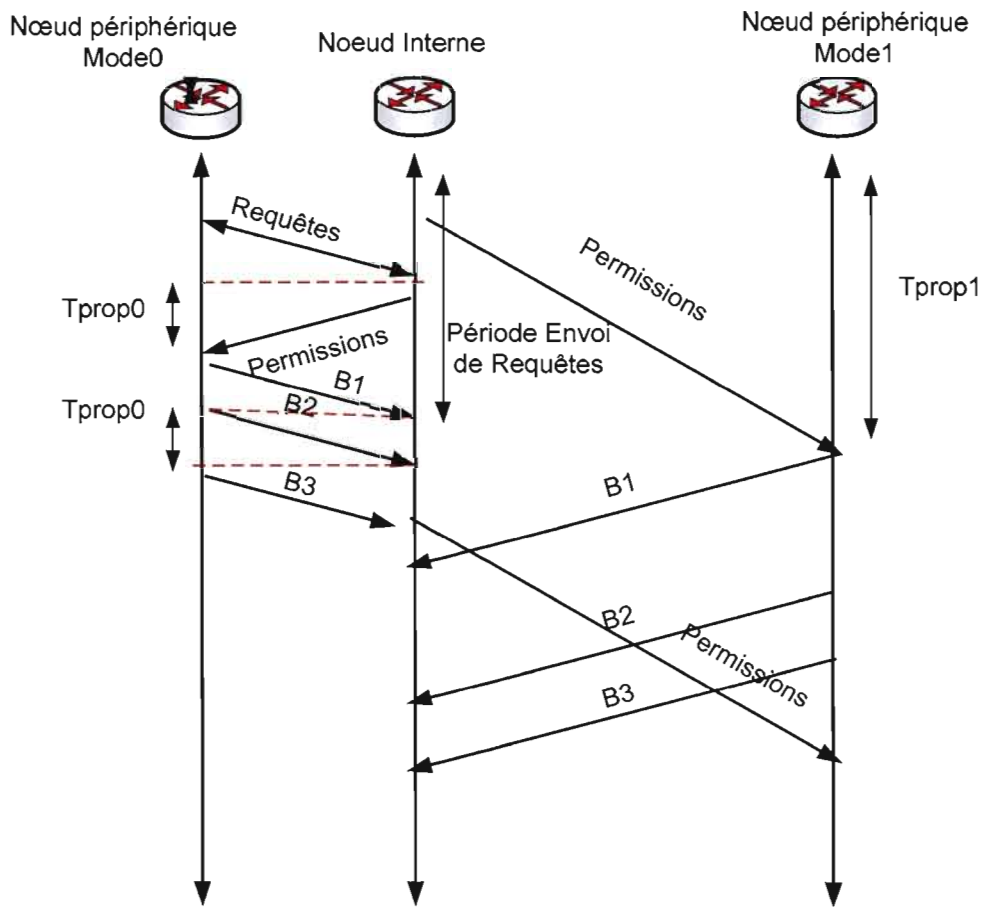
Dans [42], un schéma de signalisation bimodal a été proposé pour GOBS ; appelé aussi *Bimodal Burst Switching* (BBS). Ce schéma (Figure 15) comporte deux modes fonctionnement : Mode 0 et Mode 1. Le Mode 0 est exécuté quand le nœud périphérique est proche du nœud interne et le Mode 1 dans le cas d'un nœud distant.

Dans le Mode 0, le nœud périphérique envoie les demandes accumulées en une période fixe au nœud interne. Dès la réception de ces demandes, le nœud interne les cédulent selon Horizon et envoie une permission au nœud périphérique. Ensuite, le nœud périphérique transmet les burst dans l'ordre des permissions reçues. La période fixe pour l'envoi des demandes est courte. Ceci fournit un approvisionnement continu de permissions pour le nœud périphérique.

Cependant, dans le Mode 1, le nœud interne construit les demandes périodiquement qui sont ensuite cédulées selon horizon. La différence principale entre les deux modes est que le nœud périphérique distant (Model) n'envoi pas de demande au nœud interne. Ceci fournit un schéma plus efficace puisque le délai de propagation à sens unique est très grand. Quand le nœud périphérique (Model) reçoit une permission, il transmet les données pendant un intervalle de temps fixe.

L'architecture BBS représente une solution idéale pour les réseaux grilles puisqu'elle offre un taux de perte égale à zéro et une grande distribution géographique. Elle fournit également une bonne granularité, ce qui est un aspect important pour les applications scientifiques. Un autre avantage est que cette architecture permet une grande mise à l'échelle (*scalability*), ce qui peut fournir un partage des ressources efficace.





**Figure 15:** BBS (Bimodal Burst Switching)

### 3.3.4 WR-GOBS (*Wavelength Routed-GOBS*)

WR-OBS est une technique qui est basée sur la réservation à deux sens et qui peut offrir un service fiable et un délai de bout-en-bout plus long pour les applications de la grille [48]. Son implémentation comme technique de signalisation au sein d'un réseau grille vise donc à fournir une architecture réseau capable de satisfaire les exigences des applications nécessitant un traitement intensif et aussi de supporter les futures grilles.

Dans un scénario WR-GOBS (Figure 14), la tâche est créée au nœud d'entrée et un paquet de contrôle est envoyé au nœud de contrôle qui se chargera donc de trouver la ressource approprié

pour le traitement de la tâche. Tous les routeurs intermédiaires diffusent donc leurs services au nœud de contrôle ce qui facilite et accélère la recherche d'une destination. Ainsi, dans un réseau WR-GOBS une tâche peut trouver facilement sa destination, le chemin sera réservé pour le burst des données et un acquittement est transmis pour confirmer le traitement de la tâche. Ce dernier peut aussi être employé pour réserver la bande passante pour l'envoi du résultat.

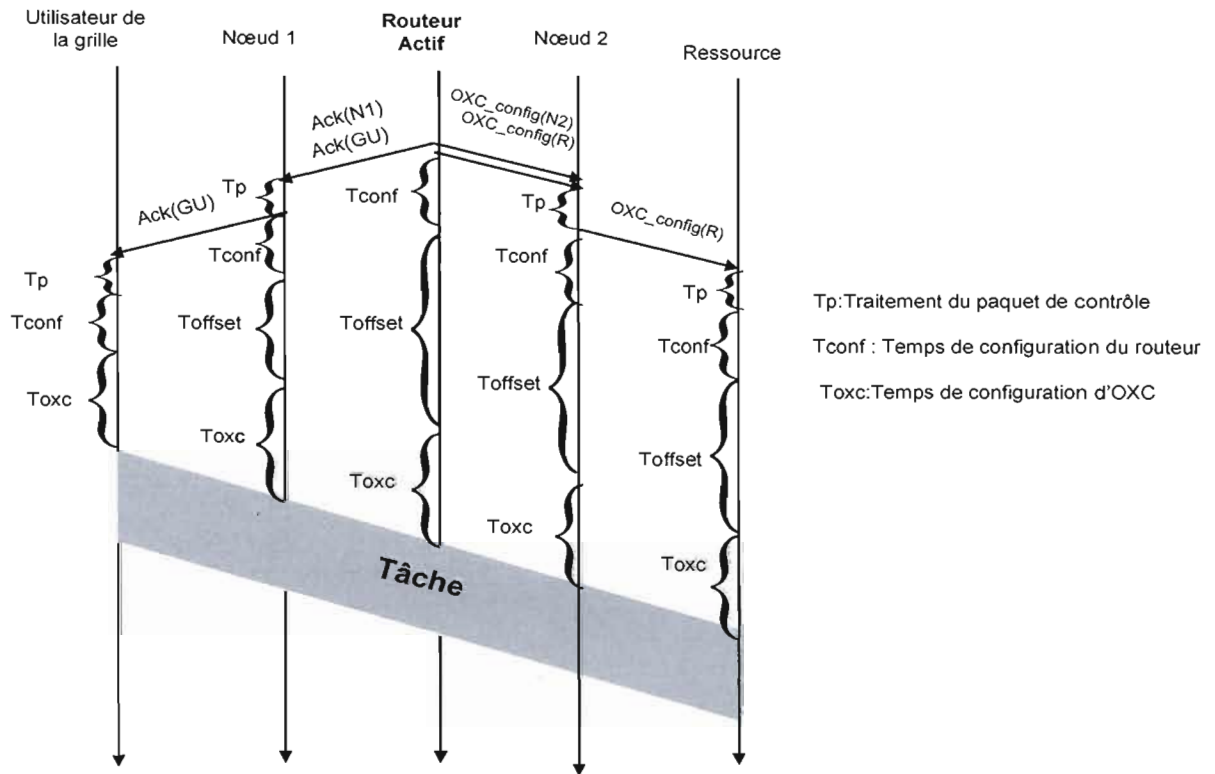


Figure 16: WR-GOBS

### 3.4 Routage Anycast:

Le processus d'un réseau GOBS est très différent d'un réseau OBS traditionnel. Par exemple, dans GOBS les tâches traitées doivent être renvoyés à leurs nœuds source (utilisateurs). En outre, un burst peut être rejeté pour au moins deux raisons : une contention au nœud intermédiaire ou insuffisance des ressources grille dans tout le réseau pour une période prédéterminé (*slack time*) [49]. Une autre différence est que les tâches non-traitées peuvent être assignées à n'importe

qu'elle adresse, tant qu'elles seront correctement traitées et retournées au nœud source. En conséquence, au lieu d'exécuter un protocole de routage unicast basé sur le court chemin où le burst doit être envoyé à un nœud spécifique, dans GOBS un protocole de routage « Anycast » peut être exécuté. Dans de tel protocole, un burst peut être envoyé à n'importe quel nœud d'OBS avec les ressources adéquates pour son traitement sinon en cas des ressources insuffisantes, les nœuds intermédiaires défléchit tout simplement le burst vers le prochain nœud approprié.

Le routage Anycast dans le cadre d'un réseau GOBS a été utilisé dans plusieurs travaux [35] [49] [53]. Un utilisateur transmet une tâche à une adresse Anycast et le réseau OBS est responsable donc de fournir une transmission *best-effort* à au moins une adresse, et de préférence à seulement une parmi les ressources disponibles.

Dans ce qui suit nous présenterons quelques algorithmes de routage Anycast proposés dans le cadre d'un réseau grille basé OBS. Ces algorithmes exécutent les deux opérations de base : l'assignation d'une destination et la déflexion. Trois algorithmes ont été définis pour l'assignation d'une destination et aussi pour la déflexion [49] [53] :

- **Soft assignation (SA) :** La source choisit un nœud destination pour le burst et qui dispose de ressources disponibles. Ce choix peut être aléatoire ou selon certaines critères. Cet algorithme peut être modifié en cas de contention ou manque de ressources. En outre, un nœud intermédiaire peut accepter un burst assigné à une autre destination s'il a les ressources suffisantes pour le traitement.
- **Hard assignation (HA) :** Cet algorithme au précédent, cependant, le nœud destination assigné par la source ne peut pas être changé par un nœud intermédiaire. HA représente principalement un routage unicast, qui est considéré un cas particulier d'Anycast.
- **Non assignation (NA) :** La source n'assigne aucune destination explicite pour le burst, elle l'envoie dans le réseau et c'est le burst qui essaye de trouver les ressources grille. Ainsi, l'éclat circulera dans le réseau jusqu'à ce qu'il trouve des ressources appropriées. Si le temps expire, l'éclat sera détruit. Quand le burst arrive à un nœud intermédiaire, le nœud vérifie ses ressources et si les ressources ne sont pas suffisantes le burst est transmis au prochain nœud sélectionné.

- **Random port availability (RPD)** : Dans le cas de contention, le burst est défléchi vers le premier port disponible sélectionné aléatoirement. Ce schéma est similaire au protocole « *hot potato* » où le nœud transmet le burst au premier canal disponible sur n'importe quel port de sortie disponible.
- **Weighted port availability (WPD)** : est similaire au RPD, sauf que le choix est basé sur une fonction de poids.
- **Weighted Grid-resource availability (WGD)** : Le nœud examine toutes les ressources disponibles dans le réseau. Puis, selon une fonction de poids, le nœud décide quel port de sortie sera choisi pour lui transmettre le burst.

En combinant les algorithmes cités ci-dessus, d'autres algorithmes (avec ou sans déflexion) ont été définis afin de comparer leurs performances au sein d'un réseau OBS et aussi pour comparer l'Anycast au routage traditionnel basé sur le plus court-chemin [49].

## CHAPITRE IV

### ARCHITCTURE DES SIMULATIONS ET RESULTATS OBTENUS

#### 4.1 Introduction

Plusieurs applications de la grille exigent le transfert d'un grand volume de données entre la source et une ou plusieurs destinations pour le stockage ou le traitement. Généralement, des ressources similaires répondant à ces exigences peuvent être situées à différents endroits dans le réseau. L'application source peut répartir ses données entre plusieurs destinations qui répondent à ses exigences. C'est le cas, par exemple, de la gestion d'une zone de stockage; dans lequel une source souhaite stocker ses données sur un ou plusieurs disques se trouvant sur différentes destinations. Ca pourrait également être le cas si l'application a besoin d'effectuer des grands calculs qui nécessitent l'utilisation de plusieurs processeurs simultanément.

Ainsi le rôle du réseau OBS, en tant qu'infrastructure de la grille, est non seulement de fournir une large bande passante, mais également de définir les ressources adéquates et assurer la communication entre une source et un ensemble de destinations [43]. La réalisation d'un réseau GOBS est fondée sur l'architecture d'OBS, ses protocoles, et quelques normes de conception afin de supporter les services de la grille.

Les demandes utilisateurs (les tâches) sont assemblées dans des bursts, qui sont transmis à travers le réseau optique vers la ressource de la grille appropriée tout en assurant les ressources réseau nécessaires pour la transmission.

Dans [51] une nouvelle infrastructure a été proposée en tirant profit du développement de nouvelles technologies dans le domaine optique. La solution proposée introduit l'utilisation des «*routeurs actifs*» au sein d'un réseau OBS et vise à fournir une structure physique capable à répondre à toutes les exigences de la grille. Plusieurs travaux se sont basés sur cette infrastructure

afin d'évaluer l'utilisation de différents protocoles de signalisation dans le cadre d'un réseau GOBS pour la réservation des ressources [44] [48] [50].

Un routeur inactif a comme principale tâche le routage des bursts à travers le réseau. Un paquet de contrôle, précédant le burst, informe le routeur à l'avance par son arrivée et le type des données que celui-ci transporte. Le routeur inactif ne fait qu'acheminer le burst vers le port approprié. Cependant, un routeur actif peut interpréter les données transportées dans le paquet de contrôle et les acheminer vers les ressources grille disponibles. Ces routeurs actifs exécutent un algorithme basé sur les spécifications de la tâche pour la découverte des ressources. Cette architecture de réseaux programmable intègre des *processeurs réseau* (*Network Processor, NP*) dans les routeurs actifs. Ces processeurs peuvent analyser les données à une très grande vitesse, permettant ainsi un traitement rapide et une transmission efficace des tâches soumis à la grille. La figure 17 illustre l'architecture d'un routeur actif.

Cette architecture permet aussi le traitement du trafic traditionnel (i.e. OBS) et le trafic de la grille simultanément. Tous les routeurs OBS exécutent la transmission des bursts quand le trafic normal transite à travers le réseau. Quand des tâches de la grille sont reçues, les routeurs actifs les prennent en charge.

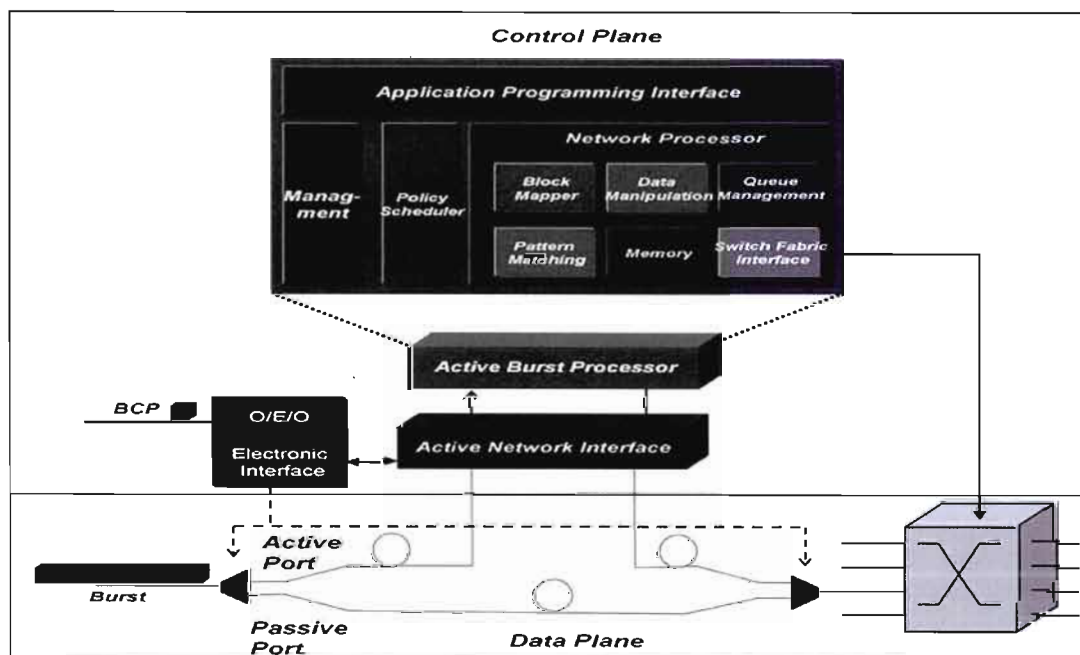


Figure 17: Architecture d'un routeur actif [50].

Dans ce qui suit, nous proposons une architecture intégrant l'utilisation des routeurs actifs et qui étudie le problème de multi-ressource au sein d'un réseau GOBS. Dans ce contexte, plusieurs ressources informatiques seront dédiées pour le traitement des tâches de la grille. Ainsi, notre objectif est de réduire au minimum le taux de perte qui résulte de la contention dans le réseau.

Nous allons en premier temps décrire le mécanisme d'allocation et réservation des ressources, ensuite nous présenterons les résultats de nos simulations faites afin d'évaluer la performance d'un réseau GOBS.

## 4.2 Mécanisme d'allocation et réservation des ressources

Dans ce travail nous employons le JET comme protocole de signalisation pour le traitement des demandes de la grille et l'assignation des ressources (schéma proposé dans [51]). Afin de l'adapter aux besoins de l'architecture active de la grille, le Jet a été modifié. Le trafic de la grille est transmis en deux étapes ce qui assurent une transmission efficace (voir Figure 18) :

- Les spécifications de la tâche sont envoyées sous forme d'un burst actif ;
- Les données à traiter (constituant la tâche utilisateur) sont transmises sous forme d'un burst inactif.

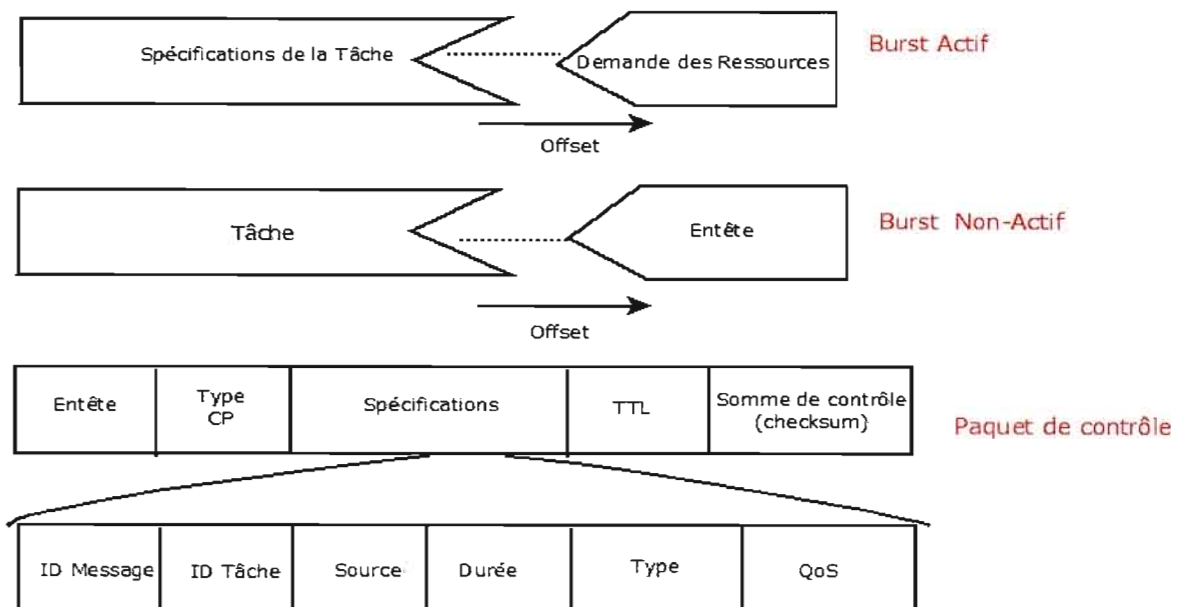
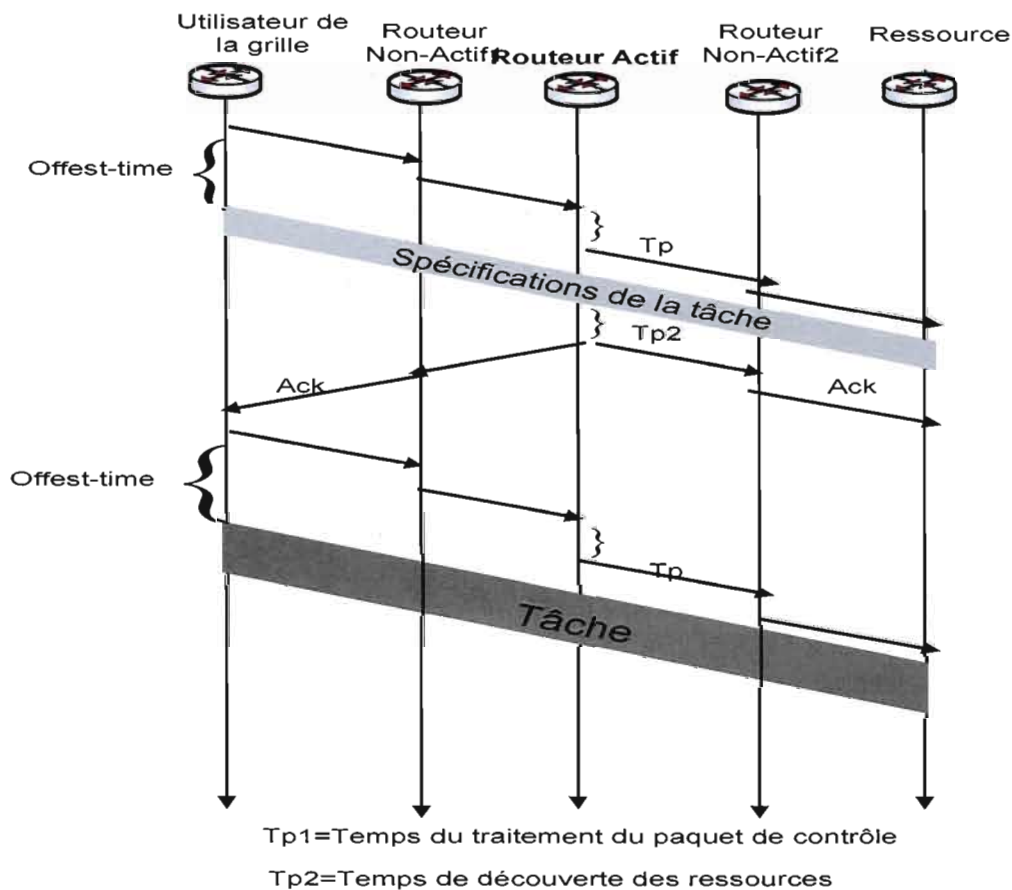


Figure 18: structure d'un burst actif et Non-actif.

L'utilisateur de la grille envoie une tâche au nœud périphérique en l'informant des ressources requises pour le traitement. Le nœud d'entrée envoie un premier paquet de contrôle pour informer les nœuds de l'arrivée d'un burst actif. Après l'*offset-time*, le burst actif est envoyé dans le réseau transportant les caractéristiques de la tâche. En recevant le burst actif, le routeur actif pourra facilement acheminer la tâche aux ressources disponibles appropriées. Le routeur périphérique est informé par un acquittement (ACK) du résultat des ressources disponibles. Il envoie ensuite la tâche à exécuter sous forme d'un burst inactif.

Cette combinaison fournit l'efficacité au niveau de la bande passante particulièrement quand un grand nombre de données doit être transféré vu que la tâche est envoyée juste quand les ressources sont réservées.



**Figure 19:** Demande de ressources et transmission de la tâche basée sur Jet



### 4.3 Simulations et résultats

Il existe peu de travaux qui se sont intéressés à l'analyse et l'évaluation globale de la performance du trafic global dans les réseaux GOBS. La plupart de ces travaux ont traité la problématique de l'architecture du point de vue des procédures de la fusion de la grille et des réseaux OBS. Rappelons que notre objectif est d'évaluer la performance du réseau GOBS en termes de taux de pertes ainsi que le délai de transmission. Nous procédons à la simulation au moyen de NS-2 en intégrant nos propres algorithmes. Nous avons considéré deux topologies réseau NSFET et une autre topologie proposée dans [54]. Ces topologies seront présentées plus loin dans ce chapitre. Pour les deux réseaux nous avons assigné un seul routeur actif pour les traitements des demandes afin de simplifier l'étude et avoir une bonne analyse. Cependant, l'utilisation de plusieurs routeurs actifs pourrait être très efficace dans le cas d'un grand réseau. Le réseau sera donc divisé en plusieurs régions actives et pour chaque région on assignera un routeur actif pour le traitement des demandes des utilisateurs.

Dans ce qui suit nous présentons les résultats obtenus après les simulations faites pour évaluer la performance d'un réseau GOBS intégrant des routeurs actifs. Nous avons considéré deux aspects :

1. Tout d'abord nous avons évalué l'emploi de routeur actif au sein d'un réseau GOBS. Cette étude nous a permis de voir l'impact de l'utilisation des deux types de trafics : actif et non actif ainsi que l'emplacement des nœuds (routeur actif et la ressource).
2. Ensuite nous nous sommes intéressés plus particulièrement au trafic actif. Dans ce cas, nous avons fait varier le nombre de ressources et la taille des tâches.

#### 4.3.1 Evaluation de la performance d'infrastructure active pour GOBS

La première étape de notre travail a été consacrée à l'évaluation de la performance d'un réseau GOBS utilisant des routeurs actifs pour la découverte et le routage des tâches de la grille. Cette première étape nous a permis d'étudier la gestion des deux trafics (actif et non actif) au sein du même réseau.

Dans cette étude nous considérons notre réseau GOBS comme une seule région active avec un seul routeur actif. Nous supposons donc que les ressources sont déjà enregistrées dans le routeur

actif d'OBS. La découverte des ressources par un utilisateur de la grille consistera donc à contacter directement le routeur actif.

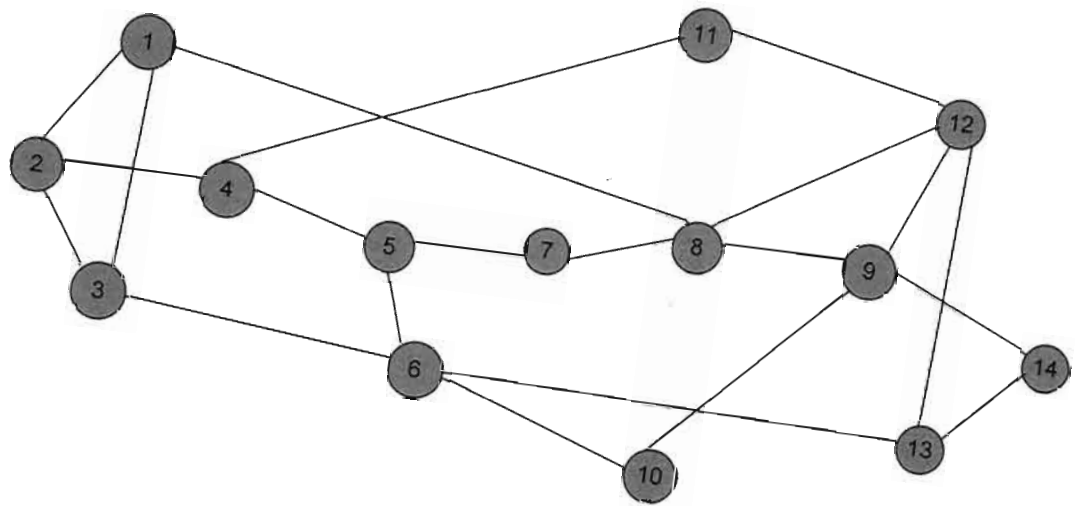
Le protocole Jet est utilisé pour la signalisation des deux trafics actif et non actif. Le trafic non-actif est envoyé dans le réseau OBS sous forme d'un burst normal. Les routeurs actifs ne le traitent donc pas. Un paquet de contrôle est envoyé à l'avance pour la réservation des ressources requises pour la transmission du burst. Après un offset-time, le burst de données est transmis dans le réseau. Nous avons fixé l'offset-time à 0.01s afin d'éviter toute congestion dû à son insuffisance.

Dans le cas du trafic actif, et comme décrit dans la section II, la transmission des tâches de la grille est divisée en deux étapes. Premièrement, un paquet de contrôle est envoyé au routeur actif pour l'informer de l'arrivée d'un burst actif. Après un offset-time, le burst est envoyé dans le réseau. Les routeurs intermédiaires non-actifs ne font donc qu'acheminer le burst vers le port approprié. Le résultat de la découverte des ressources est transmis sous forme d'un ACK (ou NACK) par le routeur actif à au nœud d'entrée. Dans le cas d'un ACK, le routeur actif informe également le nœud ressource par l'arrivée d'une tâche. Dès la réception de l'acquiescement, l'utilisateur envoie la tâche à traiter sous forme d'un burst non-actif.

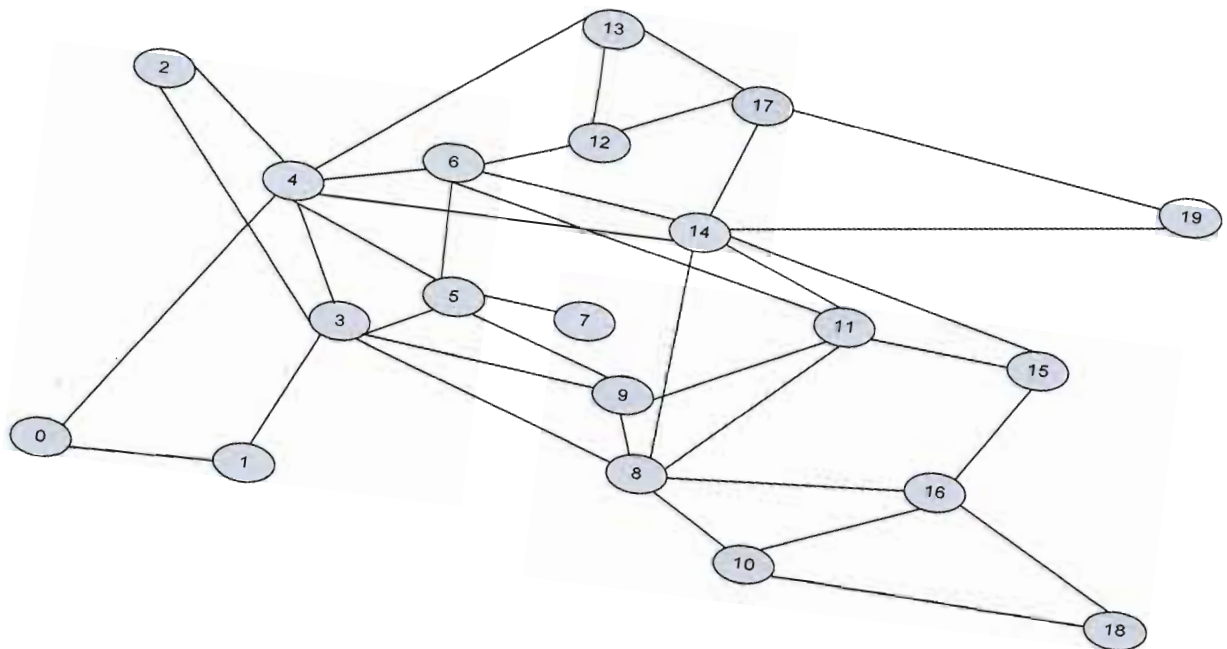
#### **4.3.1.1 Environnement de simulation**

Afin d'évaluer l'utilisation de routeur actif au sein d'un réseau GOBS et d'évaluer l'impact de son emplacement au sein du réseau, nous avons considéré deux topologies de réseau :

- La première topologie utilisée, pour l'étude de l'impact de l'utilisation du trafic actif et non actif dans le même réseau GOBS, est le réseau NSFNET (figure 20). Il est constitué de 14 nœuds et 21 liens. Le délai de propagation entre chaque deux nœud est égal à 2ms.
- La deuxième topologie a été utilisée dans [54]. Elle se compose de 20 nœuds reliés par des liens WDM bidirectionnels. Le délai de propagation entre chaque deux nœud est égal à 5ms. Tous les nœuds sont considérés comme des générateurs de tâches. Le réseau est considéré comme une seule région active avec un seul routeur actif (nœud 17). La conversion de longueur d'onde est utilisée dans les simulations. La figure 21 illustre la topologie décrite ci-dessus.



**Figure 20:** Réseau NSFNET



**Figure 21:** Topologie2

Afin de former un réseau OBS complet, un nœud d'accès est attaché à chaque nœud interne. Par défaut, tous les bursts sont routés de façon statique en utilisant le plus court chemin. Le paquet de contrôle est rejeté, si aucune ressource (réseau ou grille) ne peut être réservée. Le burst correspondant est rejeté dès son arrivée au nœud.

Les simulations ont été conduites avec les hypothèses suivantes:

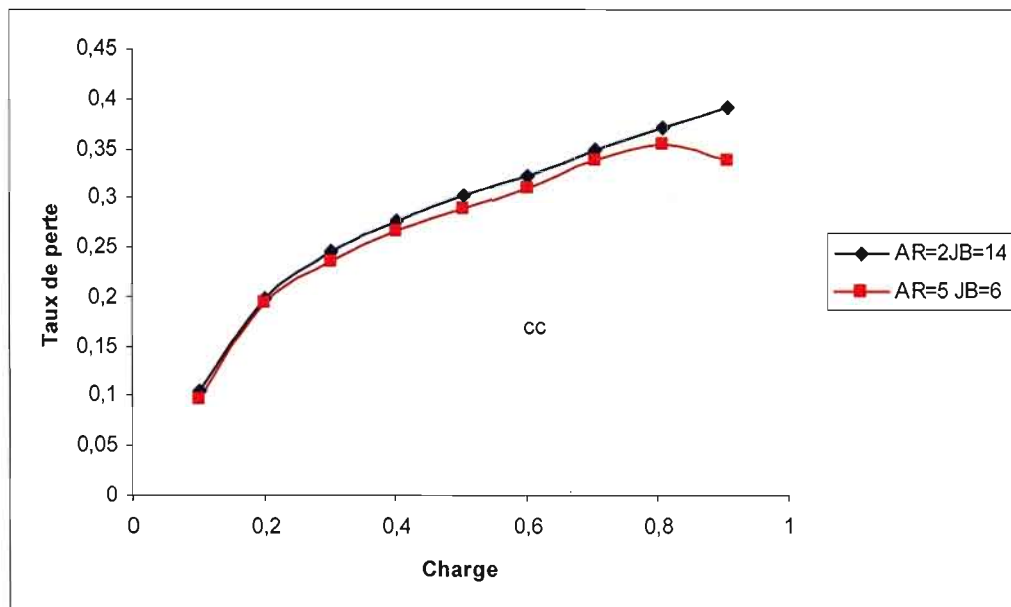
- Un générateur de trafic Poisson est utilisé.
- Un burst représente une tâche. Ceci est une hypothèse qui nous permet de simplifier le modèle de simulation.
- La taille de la tâche : 12000 octets.
- Tous les canaux ont la même bande passante: 1Gbps.
- Utilisation de la conversion de longueur d'onde.
- Chaque lien WDM a un canal de contrôle et quatre canaux de données.

#### **4.3.1.2 Résultats et analyses :**

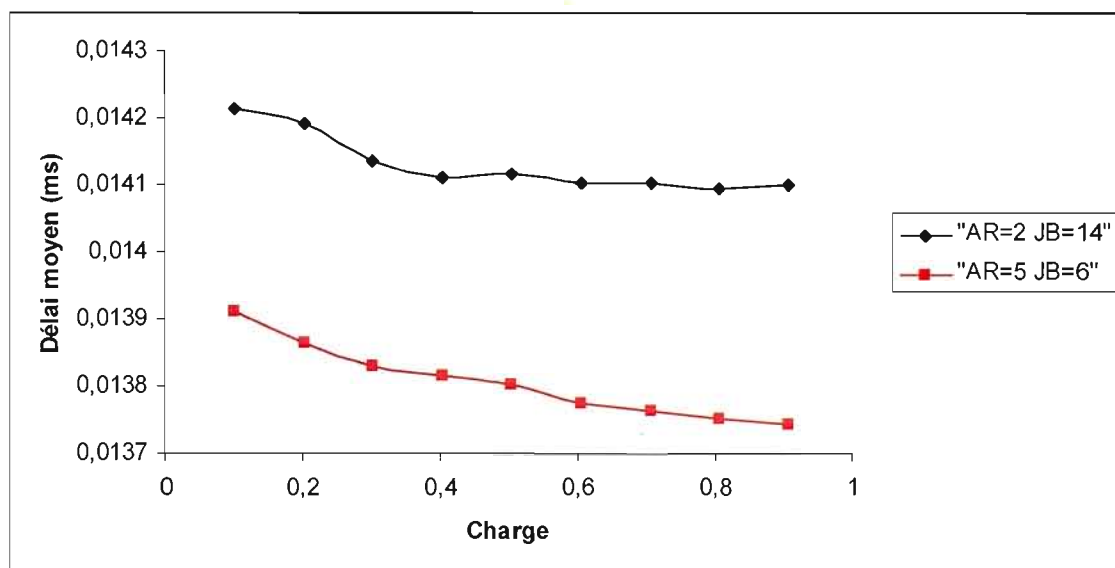
Dans cette section, nous procédons à l'analyse de performance du réseau GOBS en termes de taux de perte ainsi que de délai de transmission. Le taux de perte des bursts représente le rapport du nombre de bursts perdus sur le nombre des bursts envoyés. Le délai de transmission représente le délai moyen bout en bout. Nous comparons les résultats obtenus par simulations de deux scénarios afin d'évaluer l'impact de l'emplacement du routeur actif et le nœud ressource. Dans le premier scénario, nous avons choisis deux nœuds voisins alors que dans le deuxième scénario nous avons sélectionné le plus long chemin.

##### **NSFNET :**

La figure 22 illustre le taux de perte des bursts dans le réseau NSFNET en fonction de la charge du réseau. Nous observons que les deux scénarios donnent des résultats assez semblables. Ceci s'explique par le fait que NSFNET représente un réseau uniforme (en terme de degrés de connectivité des nœuds) ce qui donne un comportement régulier peu importe l'emplacement des nœuds. Nous avons donc tracé le délai de bout-en-bout afin de voir si l'emplacement des nœuds affecte le temps du traitement de tâches. La figure 23 représente le délai moyen en fonction de la charge. Nous observons que le délai de bout-en-bout est plus grand dans le cas du plus long chemin (routeur actif=2 et nœud ressource=14).



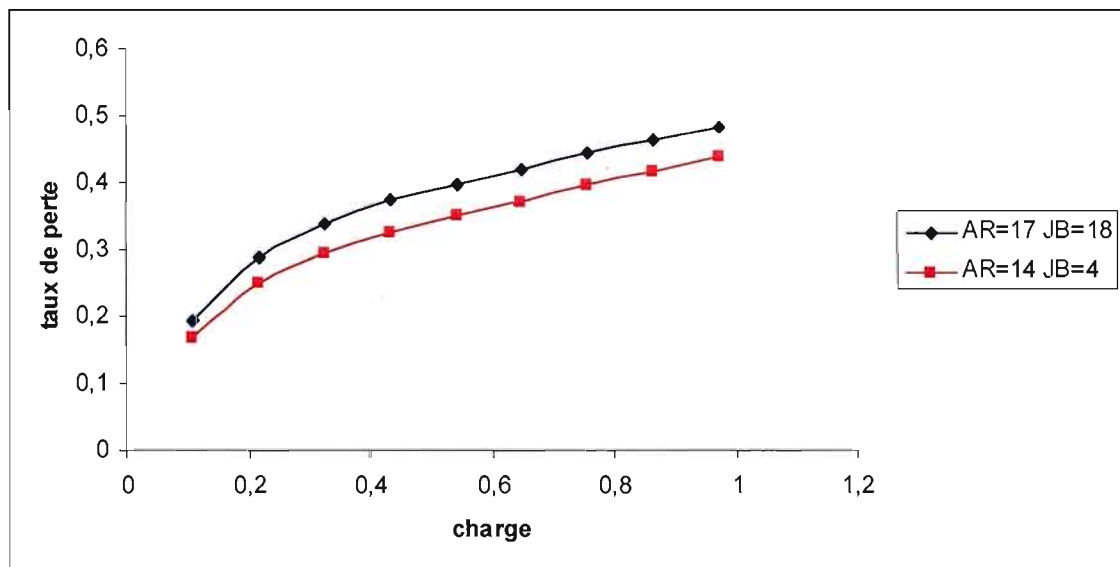
**Figure 22:** NSFNET: Taux de perte de bursts vs. Charge du réseau



**Figure 23:** NSFNET: Délai de transmission vs. Charge du réseau

### Topologie2 :

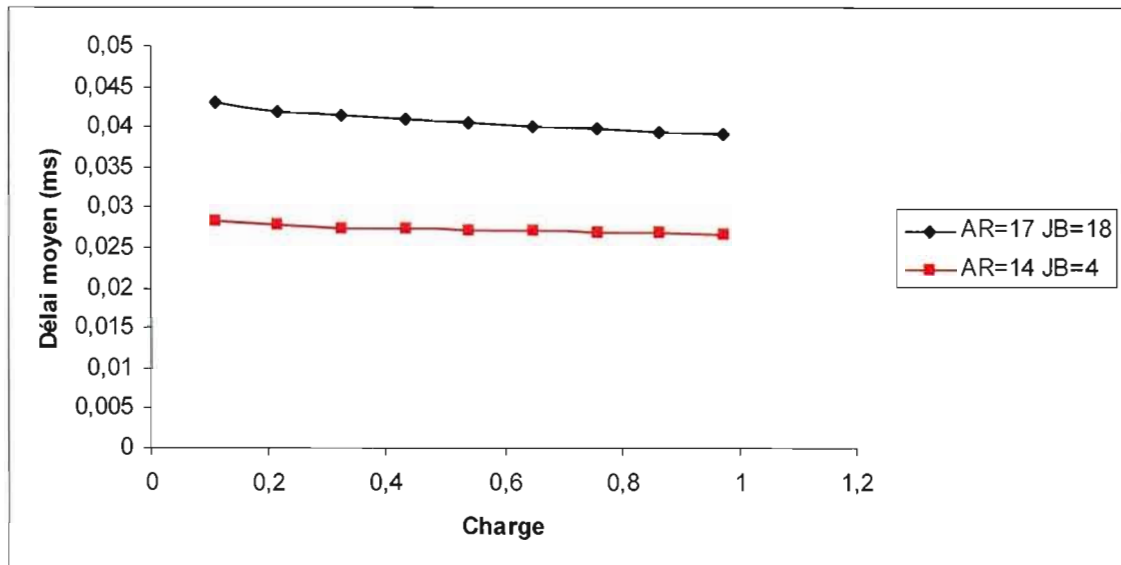
La figure 24 illustre le taux de perte des bursts en fonction de la charge du réseau. Nous avons simulé deux scénarios. Dans le premier scénario, le nœud 17 joue le rôle du routeur actif et le nœud 18 le nœud ressource (scénario 1). Pour le deuxième scénario, le nœud 14 représente le routeur actif et le nœud 4 est le nœud ressource (scénario 2). On remarque qu'avec le scénario du plus long chemin on obtient un taux de perte plus élevé. Plus la distance entre le nœud actif et le nœud ressource est grande, plus le taux de perte est élevé. Effectivement, quand le nœud ressource est loin du nœud actif, la tâche va traverser une plus longue distance avant d'atteindre la destination (le nœud ressource) et va risquer plus de contentions en traversant les nœuds intermédiaires. Par conséquent, le taux de perte enregistré depuis la source de la tâche jusqu'à la destination est élevé. Ceci nous permet de déduire que l'emplacement du routeur actif et du nœud ressource peut affecter le fonctionnement du réseau GOBS. On déduit également que le choix de l'emplacement devient de plus en plus important dans le cas d'un grand réseau et plus complexe ce qui caractérise particulièrement les grilles de calcul.



**Figure 24:** Topologie2: Taux de perte vs. Charge du réseau

La figure 25 représente le délai moyen en fonction de la charge. On observe que le temps de traitement des tâches est plus grand dans le scénario 1 (AR=17 JB=18). Ceci démontre que plus la distance entre le routeur actif et nœud ressource est grande, plus le délai de bout-en-bout est grand. Ceci nous permet de confirmer la tendance des résultats obtenus pour NSFNET, et que

l'emplacement du routeur actif et nœud ressource peut affecter le fonctionnement du réseau GOBS.

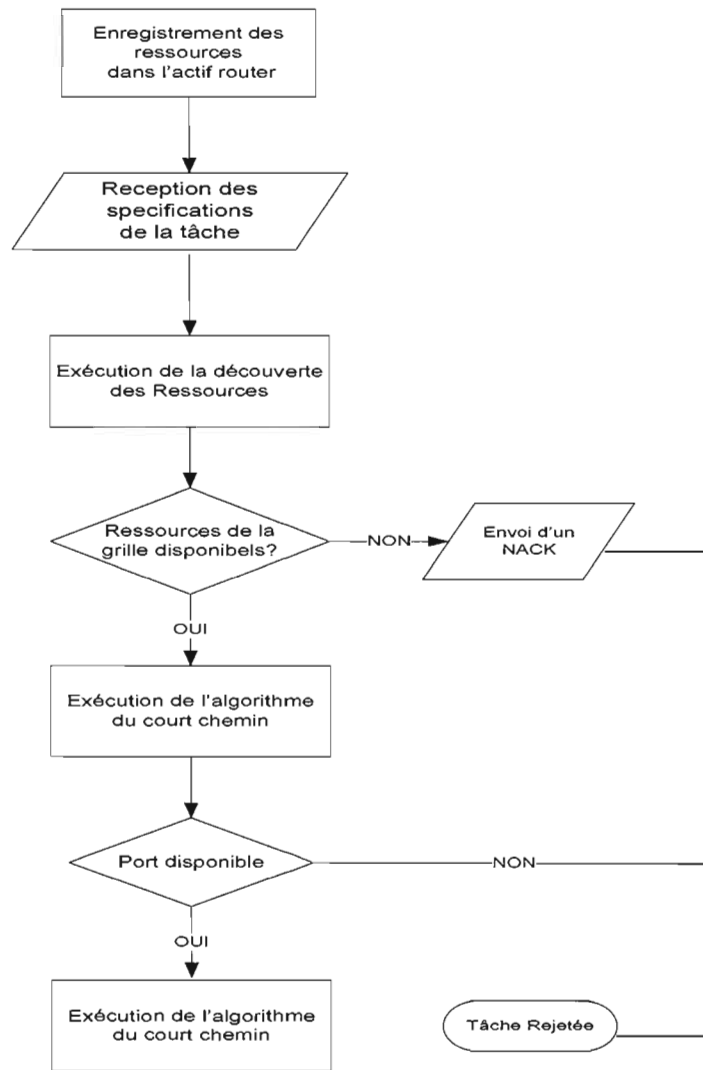


**Figure 25:** Topologie2: Délai de bout-en-bout vs. Charge du réseau

### 4.3.2 Architecture Multi-ressource pour GOBS

Dans la deuxième partie de notre travail, nous proposons une architecture multi-ressource pour un réseau GOBS. Dans ce contexte nous traitons uniquement le trafic actif (tâches de la grille). Nous utilisons la même configuration utilisée dans la première étape. Ce qui est différent, ce sera donc l'assignation de plusieurs nœuds ressources dans une même région active.

Dans ce scénario, l'utilisateur de la grille envoie une demande au routeur actif qui se charge de la découverte des ressources disponibles et qui sont déjà enregistrées dans sa table de données. Le routeur actif envoie un ACK comprenant l'adresse de la première ressource disponible. Le processus du fonctionnement est illustré dans la figure 26.



**Figure 26:** Processus de fonctionnement de GOBS

#### 4.3.2.1 Environnement de simulation

Nous utilisons les mêmes topologies que dans la section précédente. Pour évaluer la performance d'une architecture multi-ressource d'un réseau GOBS, nous mesurons le taux de perte et le taux de réussite de transmission des tâches. Le taux de réussite représente le rapport du nombre tâches reçues sur le total des tâches envoyées. Nous avons effectué plusieurs simulations en faisant varier le nombre de nœuds ressources dans le réseau. Nous avons aussi fait varier la taille de la tâche : 12000 ,9000 et 6000 octets. Cette variation va nous permettre donc de comparer le fonctionnement du réseau pour trois différentes charges vu que la charge du réseau est calculée en fonction de la taille.

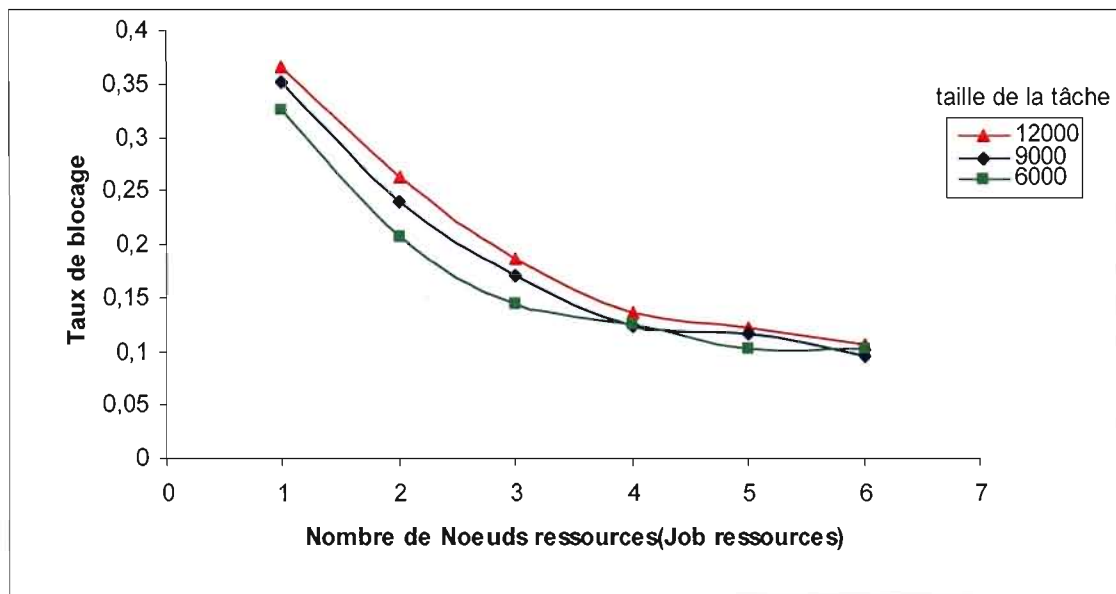


Les simulations ont été conduites avec les mêmes hypothèses considérées pour la première étape. Tous les canaux ont la même bande passante égale à 1Gbps. Pour chaque lien WDN, le nombre de canaux de contrôle est égale à 1 et le nombre de canaux de données est 4. La conversion de longueur d'onde est utilisée pour la résolution de la contention.

#### 4.3.2.2 Résultats et analyses

##### NSFNET :

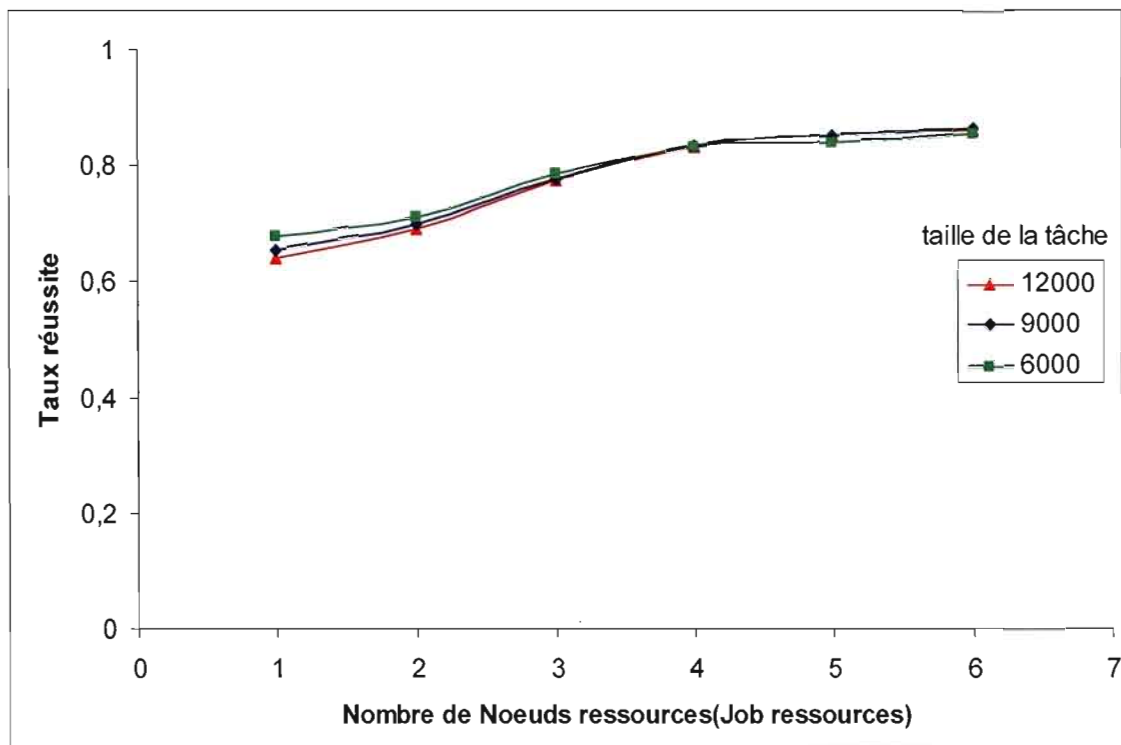
Nous considérons le nœud 11 comme routeur actif. Nous faisons varier le nombre de nœuds de ressources afin de minimiser le taux de perte. La sélection de nœuds ressources de façon à avoir le même nombre de saut entre le routeur actif et le nœud ressource. Nous nous sommes arrêtés à un maximum de 6 ce qui représente la moitié des nœuds en excluant le nœud actif. Nous observons ainsi qu'en augmentant le nombre des nœuds ressources le taux de perte diminue continuellement.



**Figure 27:** NSFNET: Taux de perte vs. Nombre de nœuds ressources

Nous avons tracé le taux de réussite en fonction de la charge afin de voir l'impact sur le traitement général des tâches. La figure 28 démontre donc qu'il y a une légère amélioration au niveau du taux de réussite mais qu'à partir de 4 nœuds ressource ce dernier se stabilise. On déduit

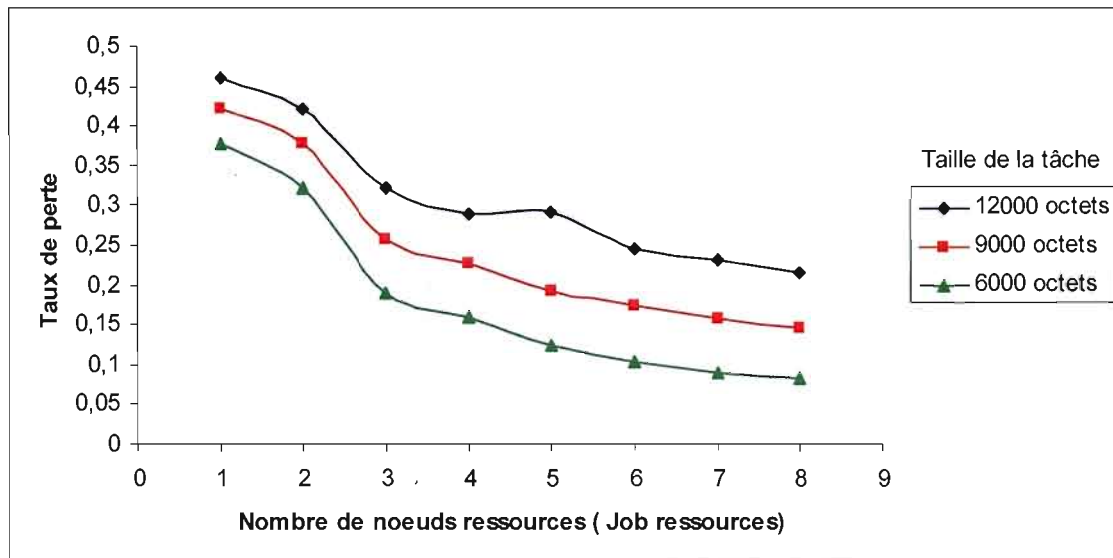
donc que le traitement des tâches n'est plus affecté par la disponibilité des nœuds ressources mais plutôt par l'indisponibilité des ressources réseau (les liens).



**Figure 28:** NSFNET: Taux de réussite vs. Nombre de nœuds ressources

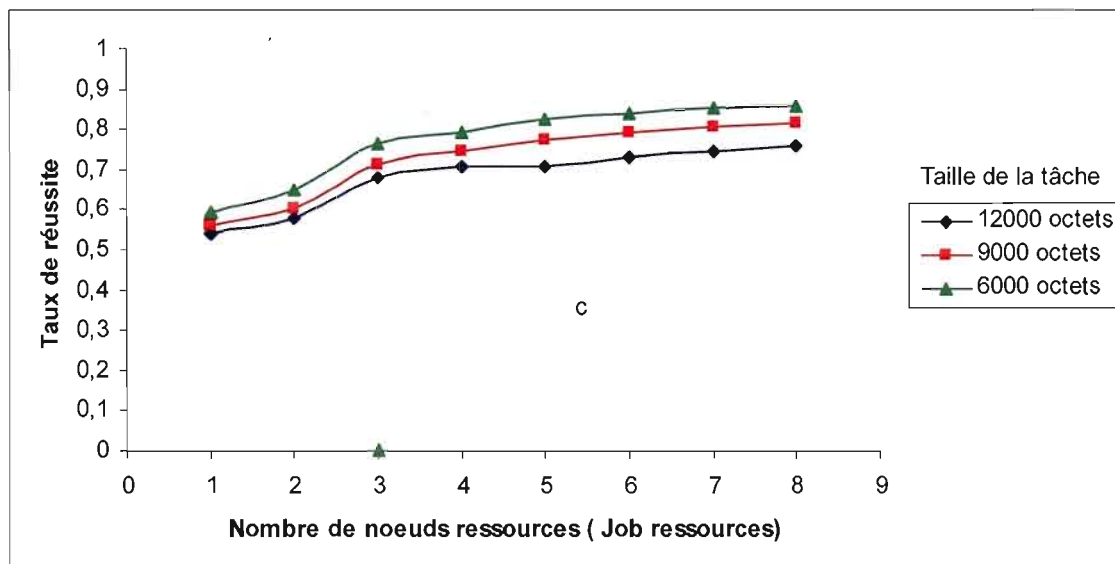
### Topologie2 :

Nous avons repris le scénario présenté précédemment où le nœud 17 représente le routeur actif et le nœud 18 est le nœud ressource. Nous avons varié le nombre de nœuds ressources afin de réduire au minimum le taux de perte. Nous avons fixé le nombre maximum de nœud ressources à la moitié des nœuds du réseau sans compter le routeur actif.



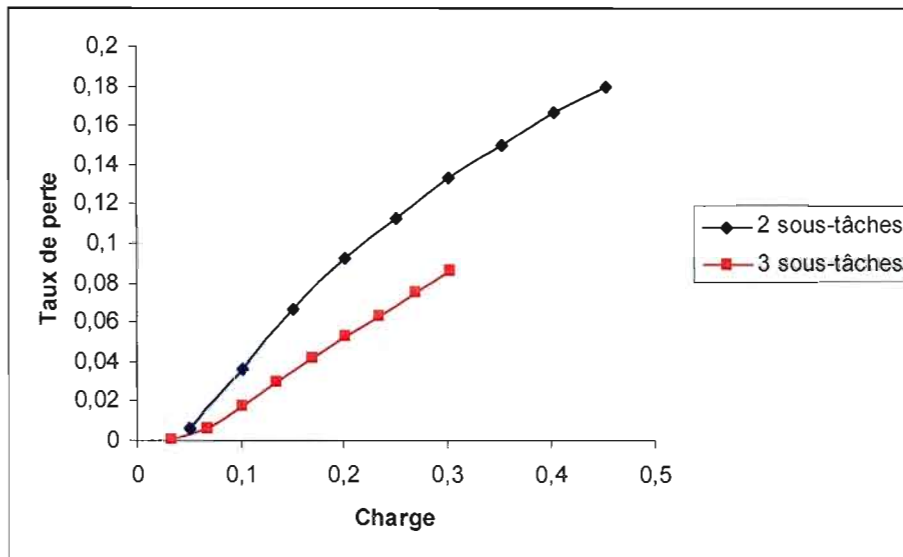
**Figure 29:** Topologie2: Taux de perte vs. Nombre de nœuds ressources

La figure 29 illustre les résultats obtenus. On observe le même comportement qu'avec NSFNET. En augmentant le nombre de nœud ressource le taux de perte diminue. Cependant et contrairement à NSFNET, on remarque que plus la taille des tâches est grande, plus le taux de perte est élevé.



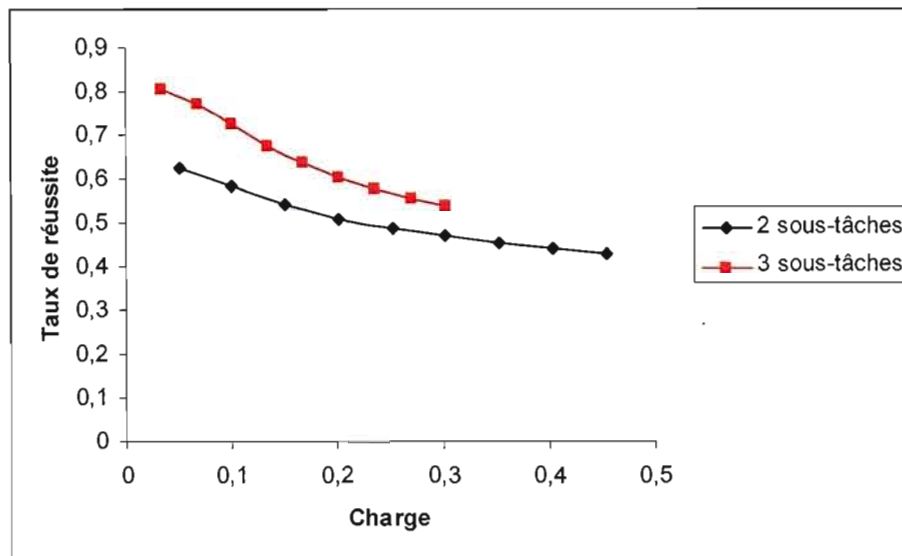
**Figure 30:** Topologie2: Taux de réussite vs. Nombre de nœuds ressources

La figure 30 nous permet d'observer qu'en assignant plusieurs nœuds ressources pour le traitement des tâches, ceci améliore le taux de réussite au sein du réseau. Il est également observé que le taux de réussite atteint jusqu'à 80% dans le cas d'une petite taille (6000 octets). Ceci nous a motivé à étudier le cas de division de la tâche en sous-tâches pour un traitement distribué. Nous supposons que la taille initiale de la tâche est 12000 octets. Premièrement, nous avons subdivisé la tâche en deux sous-tâches de 6000 octets avec deux nœuds ressources. Ensuite, nous divisons la tâche en trois sous-tâches avec trois nœuds ressources. Nous observons que la transmission des tâches sous forme de sous-tâches permet d'avoir un taux de perte moins élevé.



**Figure 31:** Taux de perte vs. Charge

Dans la figure 32 nous représentons le taux de réussite global en fonction de la charge. Le taux de réussite global représente le rapport du nombre des sous-tâches reçues sur le total des tâches envoyées. On obtient un taux de réussite plus élevé quand la tâche est divisée en trois sous tâches. Ainsi, le traitement distribué de la tâche permet d'avoir des meilleurs résultats pour une architecture multi-ressource.



**Figure 32:** Taux de réussite global vs. Charge

## Conclusion

Dans ce travail nous avons étudié l'architecture et l'optimisation de gestion des ressources d'un réseau grille basée sur OBS. Le principe grille de calcul est de découper une application lourde en petites tâches isolées, confiées à des ordinateurs différents à travers le réseau. Cette nouvelle infrastructure permet d'exploiter efficacement la puissance de calcul et les espaces de stockage inutilisés des ordinateurs.

Comme discuté dans le chapitre 3, OBS représente l'infrastructure idéale pour les grilles actuelles et futures. La motivation principale est la capacité de cette technologie à offrir une gestion de réseau à efficacité élevée et à un coût raisonnable. Un autre avantage de cette combinaison est l'aptitude de supporter un grand nombre d'applications.

Nous avons étudié et évalué dans ce mémoire la performance d'un réseau grille basé sur OBS (GOBS). Cette étude nous a permis de voir de l'impact de l'utilisation des deux types de trafics : actif et non actif ainsi que l'emplacement des nœuds (routeur actif et la ressource).

Nous avons présenté aussi le problème de multi-ressource où on assigne plusieurs ressources pour le traitement des tâches de la grille. Cette approche est efficace pour le traitement distribué au sein de la grille et particulièrement dans le cas de grandes tâches. Elle permet de réduire au minimum le taux de perte et aussi assurer une performance en termes de temps.

L'ensemble des travaux futurs se concentreront sur l'ordonnancement (Scheduling) des tâches au sein de la grille vers les ressources appropriées. Aussi un autre développement futur sur ce projet, est d'évaluer la performance en même temps au niveau des ressources grille et les ressources réseaux.

## Références

- [1] Cao X, Li J, Chen Y et Qiao C, 2000, « Assembling TCP/IP Packets in Optical Burst Switched Networks », IEEE GLOBECOM02.
- [2] Vokkarane V.M, Haridoss K, et Jue J.P, 2002, « Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks », Proceedings », SPIE OptiComm, Boston, MA, vol.4874, pp. 125-136.
- [3] Luo J, Zeng Qi, Chi. H, Zhang Z, Zhao H, 2003, « The Impacts of Burst Assembly on the Traffic Properties in Optical Burst Switching networks », Proceedings of ICCT.
- [4] Thysebaert P, Volckaert B, De Leenheer M, De Turck F, Dhoedt B, Demeester P, 2005 , « Resource Partitioning Algorithms in a Programmable Service Grid Architecture », Lecture Notes in Computer Science 3516, pp. 250-258
- [5] Volckaert B, Thysebaert P, De Leenheer M, De Turck F, Dhoedt B, Demeester P, 2004, « Grid Computing: The Next Network Challenge », The Journal of The Communications Network, Volume 3 Part 3.
- [6] Hudek G.C, Muder D.J, 1995, « Signaling analysis for a multi-switch all-optical network » Communications », IEEE International Conference.
- [7] Yao M., et C. Qiao, 1997, « Just-Enough-Time (JET): A High Speed Protocol for Bursty Traffic in Optical Networks », IEEE/LEOS Conf. on Technologies for a Global Information Infrastructure,.
- [8] Michael Düser and Polina Bayvel, 2001, « Bandwidth Utilisation and Wavelength Re-Use in WDM Optical Burst-Switched Packet Networks », *Presented at IFIP 5th Working-Conference on Optical Network Design and Modelling*.
- [9] Kirci, P., Zaim, A.H , 2006 , « Comparison of OBS Protocols », Computer Networks 2006 International Symposium on,
- [10] Wei J.Y, McFarland R.I Jr, 2000, « Just-In-Time Signaling for WDM Optical Burst Switching Networks », Journal of Lightwave Technology, vol. 18, no. 12.
- [11] Teng J and Rouskas .G, 2005, « A Detailed Analysis and Performance Comparison of Wavelength Reservation Schemes for Optical Burst Switched Networks », Photonic Network Communications.
- [12] Turner Jonathan S, 1999, « Terabit Burst Switching Progress Report », Journal of High Speed Networks.
- [13] Qiao C, Yoo M, 1999, « Optical Burst Switching (OBS): A New Paradigm for an Optical Internet », Journal of High Speed Networks.

- [14] Vinod M. Vokkarane, 2003, « Prioritized Burst Segmentation and Composite Burst Assembly Techniques for QoS Support in Optical Burst-Switched Network », *IEEE Journal on selected areas in communications*, vol. 21, no. 7.
- [15] Rosberg Z, Zalesky A, Vu H.L, 2006, « Analysis of OBS Networks With Limited Wavelength Conversion », *IEEE/ACM Transactions on Networking*, vol. 14, no. 5.
- [16] Jing Teng , 2005, « Wavelength Selection in OBS Networks Using Traffic Engineering and Priority-based Concepts », *IEEE journal on selected areas in communications*, vol. 23, no. 8.
- [17] Kim B.C, Cho Y, Lee J, Choi Y, and Montgomery D, 2002, « Performance of Optical Burst Switching Techniques in Multi-Hop Networks », *Global Telecommunications Conference GLOBECOM '02*.
- [18] Sanjeev Verma, Hemant Chaskar, and Rayadurgam Ravikanth, 2000, « Optical Burst Switching: A Viable Solution for Terabit IP Backbone », *IEEE Network*.
- [19] Jeong H, Cho J, Kang M, Park H, Choi J, Won Y, 2002, « GMPLS over OBS Architecture for an Optical Packet Internet », *COIN-PS 2002*, pp. 84-86,
- [20] Yuhua Chen, 2007, « Optimal Burst Scheduling in Optical Burst Switched Networks » *journal of lightwave technology*, vol. 25, no. 8.
- [21] Xu J, Qiao C, Li J, et Xu G, 2003, « Efficient Channel Scheduling Algorithms in Optical Burst Switched Networks », *IEEE INFOCOM*.
- [22] Michael Düser, 2002, « Analysis of a Dynamically Wavelength-Routed Optical Burst Switched Network architecture », *Journal of Lightwave Technology*, vol. 20, no. 4.
- [23] Ian Foster, 2001, « The Anatomy of the Grid Enabling Scalable Virtual Organizations », *Proceedings of the 1st International Symposium on Cluster Computing and the Grid (CCGRID '01)*.
- [24] De Leenheer M, Van Breusegem E, Thysebaert P, B. Volckaert, De Turck F, Dhoedt B, Demeester P, Simeonidou D, O' Mahoney M. J, Nejabati R, Tzanakaki A, Tomkos I, 2004, « An OBS-based Grid Architecture », *IEEE Communications Society, Globecom Workshops*.
- [25] Hernandez Fabio, Jacq Nicolas, Nicoud Sophie, « DataGrid projet Européen de Grille de calcul ».
- [26] Site officiel du projet DataGrid: <http://www.eu-datagrid.org>
- [27] EuroGrid : <http://www.eurogrid.org>
- [28] Acampora A. S. et Shah I. A., 1992, « Multihop Lightwave Networks: A Comparison of Store-and-Forward and Hot-Potato Routing », *IEEE Transactions on Communications*, vol. 40, no. 6, pp. 1082-1090.
- [29] Bononi A, Castanon G.A et. Tonguz O. K, 1999, « Analysis of Hot-Potato Optical Networks with Wavelength Conversion », *IEEE Journal of Lightwave Technology*, vol. 17, no. 4.



- [30] Forghieri F, Bononi A et Prucnal P. R, 1995, « Analysis and Comparison of Hot-Potato and Single-Buffer Deflection Routing in Very High Bit Rate Optical Mesh Networks », IEEE Transactions on Communications, vol. 43, no. 1.
- [31] Ramaswami R et. Sivarajan K.N, 1995, « Routing and Wavelength Assignment in All-Optical Networks », IEEE/ACM Transactions on Networking, vol. 3, no. 5.
- [32] Chlamtac I, Fumagalli A, G. Kazovsky, et al, 1996, « CORD: Contention Resolution by Delay Lines », IEEE Journal on Selected Areas in Communications, vol. 14, no. 5.
- [33] Yao S, Mukherjee B, Yoo S.J.B et Dixit S, 2000, « All-Optical Packet-Switched Networks: A Study of Contention Resolution Schemes in an Irregular Mesh Network with Variable-Sized Packets », Proceedings SPIE OptiComm 2000, Dallas, TX, pp.235-246.
- [34] Vokkarane V, Jue J.P and Sitaraman S, 2002, « Burst Segmentation: an Approach for Reducing Packet Loss in Optical Burst Switched Networks », Proceedings, IEEE International Conference on Conference (ICC) 2002, NY, vol. 5, pp. 2673-2677.
- [35] De Leenheer M, Develder C, Stevens T, Dhoedt B, Pickavet M, Demeester P, 2007, « Design and Control of Optical Grid Networks », Broadband Communications Networks and Systems.
- [36] Wu R, Ji Y, 2006, « An Aware-QoS Network Resource Parallel Scheduling Scheme in OBS-Based Photonic Grid », Optical Internet and Next Generation Network, COIN-NGNCON 2006, The Joint International Conference.
- [37] Biswanath Mukherjee, 2000, « WDM Optical Communication Networks: Progress and Challenges », Ieee Journal on Selected Areas in Communications, vol. 18, no. 10.
- [38] Modiano E, 1999, « WDM-Based Packet Networks », Communications Magazine, IEEE, Volume: 37, p 130-135.
- [39] Simeonidou D, Nejabati R, O'Mahony M, 2004, « Photonic Infrastructure for Grid Enabled Networks », IEEE International Symposium on Cluster Computing and the Grid.
- [40] Simeonidou D, Nejabati R, O'Mahony M, 2004, « An Optical Network Infrastructure Suitable For Global Grid Computing », TERENA Networking Conference.
- [41] Van Breusegem E, De Leenheer M et al, 2004, « An OBS architecture for pervasive grid computing », Workshop on Optical Burst Switching, BROADNETS.
- [42] Mountroudou X, Perros H.G, Beshai M, 2005, « Performance evaluation of an optical burst switching scheme for grid networks », Broadband Networks 2nd International Conference .
- [43] Nejabati R, 2007, « Grid Optical Burst Switched Networks (GOBS) », Grid High-Performance Networking Group (GHPN), Open Grid Forum (OGF).
- [44] De Leenheer M, Thysebaert P, Volckaert B, De Turck F, Dhoedt B,. Demeester P, Simeonidou D, Nejabati R, Zervas G, Klonidis D, O'Mahony M.J, 2006, « A View on Enabling Consumer Oriented Grids through Optical Burst Switching », IEEE communications Magazine, 44(3):124-131.

- [45] Kumar V. P et al , 1998, « Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet », IEEE Commun. Mag, vol. 36, pp.152–64.
- [46] Byungsang Kim; Ji-Hwan Kim; Min-Gon Kim; Minho Kang; « Application-aware Contention Resolution Scheme for Grid Services over OBS Networks » , 2006, Advanced Communication Technology, The 8th International Conference, Volume 2.
- [47] Min Li; Xueyue Cao, Leiming Xu, Ping Zhang; Chi Yuan, Shuping Peng, Zhengbin Li, 2008 « A Distributed Fast Light-path Set-up Protocol for Grid over OBS Networks », IFIP International Conference .
- [48] Avo R, Guerreiro M, Correia N.S.C, Medeiros M.C.R, 2007, « A Signaling Architecture for Consumer Oriented Grids Based on Optical Burst Switching », Networking and Services 2007.ICNS Third International Conference.
- [49] Farahmand F, De Leenheer M, Thysebaert P, Volckaert B, De Turck F, Dhoedt B, Demeester P, Jue J.P, 2005, « A Multi-Layered Approach to Optical Burst-Switched Based Grids », Proc. Fifth International Workshop on Optical Burst/Package Switching (WOBS).
- [50] Simeonidou D, Nejabati R, Zervas G, Klonidis D, Tzanakaki A et O'Mahony M.J, 2005, « Dynamic Optical Network Architectures and Technologies for Existing and Emerging Grid Services », Journal of Lightwave Technology.
- [51] Nejabati R, Zervas G, Dimitriades G et D. Simeonidou , 2005, « Programmable Optical Burst Switched Network: A Novel Infrastructure for Grid Services », In Proc. Fifth IEEE International Symposium on Cluster Computing and theGrid (CCGrid), volume 2, p993–999.
- [52] Nejabati R, Zervas G, Simeonidou, D, 2008, « Novel implementations and test-bed demonstrations of application-aware OBS network architectures », Transparent Optical Networks, ICTON 2008, 10th Anniversary International Conference on Volume 3.
- [53] De Leenheer M , Farahmand F, Kejie Lu, Tao Zhang, Thysebaert P, De Turck F, Dhoedt B, Demeester P, Jue J.P, 2006, « AnyCast Algorithms Supporting Optical Burst Switched Grid Networks », Networking and Services, ICNS '06 International conference .
- [54] Guerreiro M, Correia, N.S, Medeiros M.C.R, 2007, « Evaluation and Comparison of Signaling Reservation Protocols for Grid over OBS Networks Employing Active Routers », Transparent Optical Networks, OCTON '07 , 9th International Conference on Volume 3.